
information
math
potential sources information
non rational facts
fact
source
designer
program
use
effects
building
documentation
schematics
development
contract docum.
values
whole
parts
program outline design
proj 1
proj 2
designer
understanding
building
stages of synthesis
could be
project 1
project 2
project 3
feedback
evaluation
contents
continuity
schematics
program

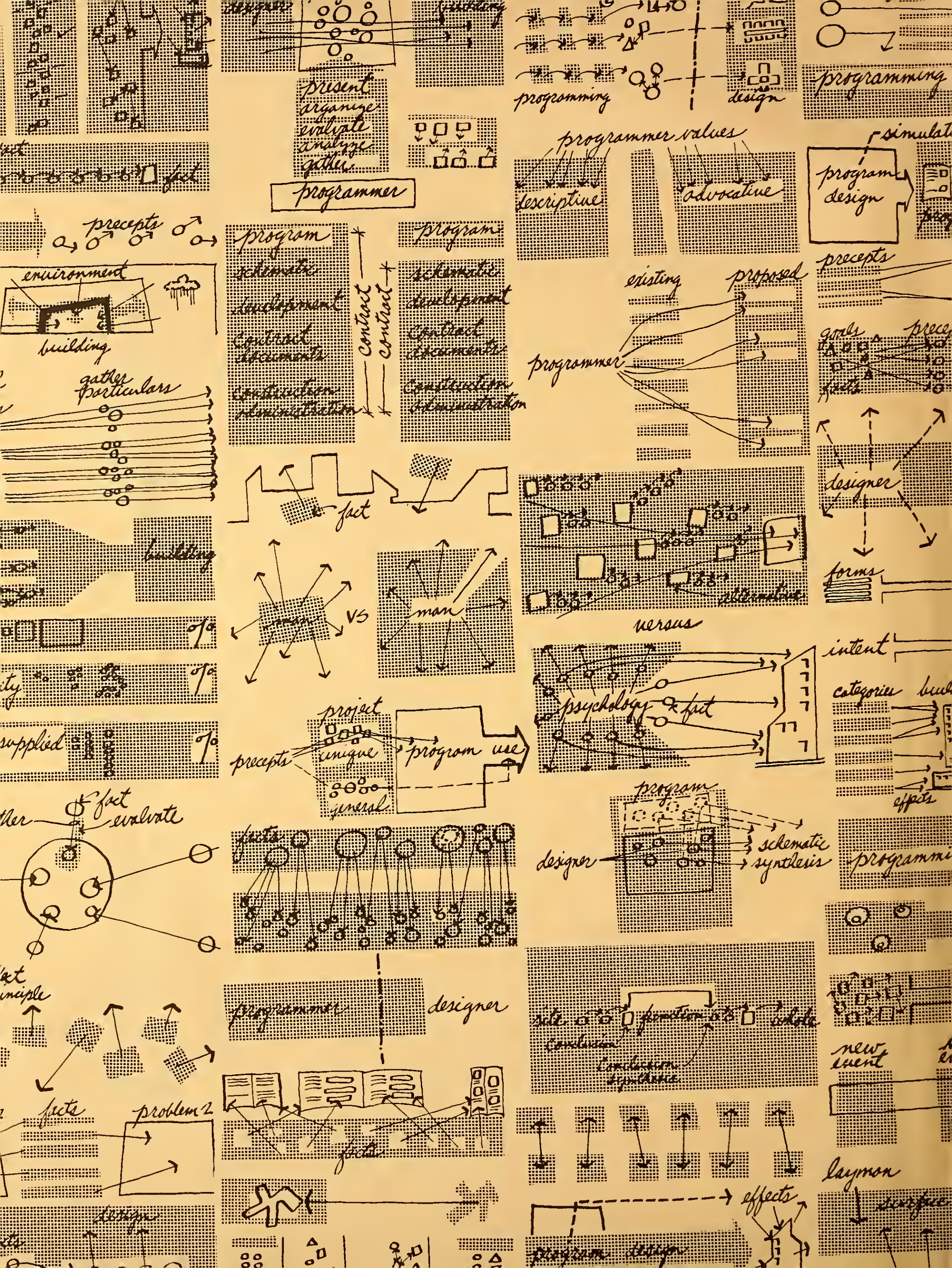
5.00 ~~9.50~~ Book
1.00 ~~1.00~~ Postage and Handling
6.00 ~~10.50~~ Total Prepaid

Architectural Media
P. O. Box 4664
Tucson, Arizona 85717

INTRODUCTION TO ARCHITECTURAL PROGRAMMING

EDWARD T. WHITE ■ ARCHITECTURE ■ UNIVERSITY OF ARIZONA

TABLE OF CONTENTS



present
organize
evaluate
analyze
gather
programmer
programming
design
programmer values
descriptive
advocative
program design
precepts
environment
building
program
schematic
development
contract documents
construction administration
contract
existing
proposed
goals
facts
gather particulars
fact
designer
forms
man
vs
versus
psychology
intent
categories
effects
project
unique
general
program use
evaluate
schematic synthesis
programmer
problem 2
conclusion
synthesis
new event
layman

INTRODUCTION 2

PREFACE

INTENT
SCOPE
ORGANIZATION AND FORMAT

PROGRAMMING PARADIGM

MODELS
RELATIONSHIPS : VIEW OF DESIGN TO PROGRAMMING
DEVELOPING A VIEW OF DESIGN
PROGRAMMING - DESIGN MODEL

PREFACE

Although its FORM and ROLE may vary from project to project and from design method to design method, PROGRAMMING is nevertheless an integral part of the planning of any building. With the architect involved in projects of greater and greater complexity, the value of the program has grown from a means of "getting to know the problem" to that of an instrument which LIMITS and DIRECTS the planning process. Whereas in the past programming amounted to little more than a superficial involvement with familiar and uncomplicated functions which had little or no direct influence on the operations of design synthesis, it is developing into a systematic, analytical discipline with ever increasing INTERFACE with planning operations. The increasing number of firms which specialize in this area is evidence of the new importance placed on programming and its recognition as a distinct component of the design process.

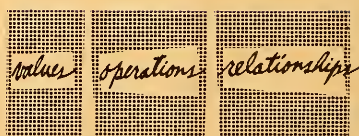
I. INTENT

A. This book is meant to:

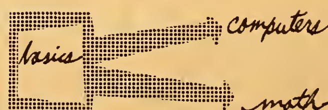
1. PROMOTE the concept and value of programming.
2. SERVE as a text for introductory programming courses.
3. AID the practitioner as a guide in developing his own programming services.
4. PROVIDE clients with a general introduction to programming as a needed service in facilities planning.

II. SCOPE

A. Emphasis will be placed on the VALUE of the different aspects of programming, the OPERATIONS involved in writing and responding to a program, and the RELATIONSHIPS between issues within programming, between programming and design synthesis, and between program and the final design.

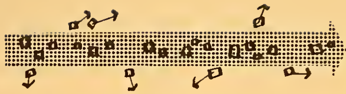


B. Only TRADITIONAL architectural programming operations are discussed. There is no treatment of mathematical models or computer applications. The use of these more sophisticated techniques first demands development of a clear understanding of BASIC programming concepts.

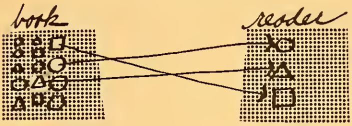


C. The contents offer an INTRODUCTORY overview of programming as an architectural activity. The book does not claim to comprehensively cover ALL aspects and attitudes

of the field. There are many TANGENTIAL issues which have not been pursued because of the introductory nature of the book.



D. Although there is an inevitable PERSONAL view of design and programming which has served to provide the basic organization of the contents, there has been an effort to present the information in a way that facilitates the reader's assembly of HIS OWN programming paradigm.



E. The subject matter includes both THEORETICAL and PRACTICAL aspects of programming.

III. ORGANIZATION AND FORMAT

A. The book is divided into INTRODUCTORY issues, BACKGROUND concerns which provide a context for discussing programming and considerations that apply directly to the PROGRAMMING operations.

B. The text is in OUTLINE form with accompanying explanatory diagrams where appropriate.

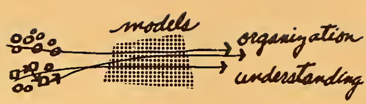
C. A table of SUB-CONTENTS occurs at the start of each of the three major divisions.

D. The subject matter is organized around the PROGRAMMING PARADIGM presented below.

5

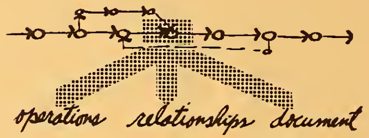
PROGRAMMING PARADIGM

I. MODELS



A. Where there are complex operations to be performed or a large body of information to be presented, the use of MODELS often proves useful.

Models or paradigms provide a WAY of understanding information or operations and their relationships and so also serve as MEANS for organizing and presenting ideas about both.



B. The programmer's VIEW OF DESIGN as a process often helps to establish the ROLE of programming in that process. Role in turn assists in the determination of specific OPERATIONS and RELATIONSHIPS and in establishing the NATURE of the programming document.

II. RELATIONSHIPS: VIEW OF DESIGN TO PROGRAMMING

A. The OPERATIONS performed and their sequence in design are largely a result of the designer's PERSONAL attitude and values.

B. As PROGRAMMING is part of the DESIGN process, it is reasonable to assume that the designer's view of design will influence the programming phase just as any other phase. If the designer is not the programmer, he is nevertheless often in a position to set the goals of the program and so, in effect, direct its operations and final form.

C. Consistency in values regarding programming and design synthesis is vital to insuring a SMOOTH TRANSITION from problem statement to solution. If a program is written with a different view of design than the designer has, he may have difficulty relating to it in trying to solve the problem.

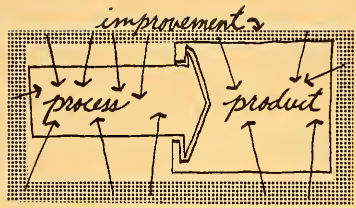
D. In order to insure this consistency, the designer must be aware of his ATTITUDES and VALUES about design. The more complete this awareness in this regard, the more able he will be to tailor the programming phase to his particular design problem.

III. DEVELOPING A VIEW OF DESIGN

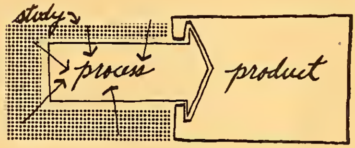
A. In all professions there is not only a concern for the quality

of the PRODUCT but also a value placed on the quality of the PROCESS that produced it.

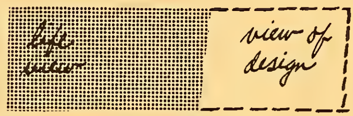
B. In architectural design this means it is important to not only arrive at a good BUILDING DESIGN but also continually work to improve the PROCESS for ARRIVING at solutions. This requires that an attempt be made to bring as much of the process to CONSCIOUS AWARENESS as possible. It also requires an analysis of values and attitudes with respect to major design PROCESS ISSUES even though in time they may EVOLVE and CHANGE.



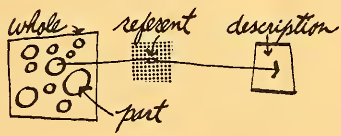
C. This self analysis to arrive at a "view of design" cannot occur while "doing a design." When attention is focussed on MAKING a product it cannot also be focussed on the PROCESS of production. This demands a "stepping back," as it were, and reflecting upon what is done when designing. What kinds of things in design are VALUED?



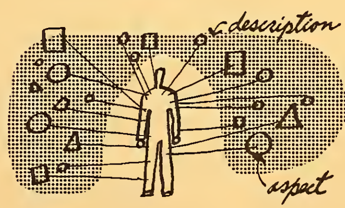
D. A view of DESIGN is an extension of a broader LIFE view. In reflecting on our view of design, sometimes we discover something about our value system in general. In the same way, an awareness of our values on broader issues can be of help in analyzing our view of design.



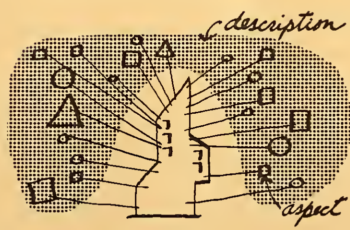
E. Descriptions always involve the COMPRISING COMPONENTS of what we are describing and their RELATIONSHIPS to other things we know. Our knowledge of something is more complete the more we become aware of its relationships or view it from DIFFERENT STANDPOINTS.



F. For example, to attempt to know a person better demands that we know how he acts in different circumstances (talking, acting under stress, tendencies when depressed, tendencies when content) and what his views are with respect to given issues (foreign policy, civil rights, euthanasia, women's lib., abortion, politics). It is the accumulation of all these INDIVIDUAL and SPECIFIC items that result in KNOWING or describing the person.



G. Another example is knowing or describing a building. It is impossible to describe it as a WHOLE. Only through the accumulation of specific individual ASPECTS about the building can it be described or known (structural system, mechanical concept, form, light patterns, geometry, response to context). In fact, even these categories are too broad to describe as WHOLES and would need to make reference to COMPONENTS within themselves in order to arrive at an adequate description.



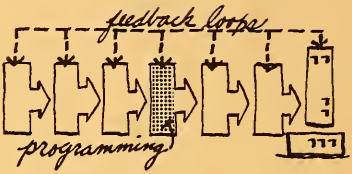
H. Our "view of design" is a result of our values and attitudes

with respect to many INDIVIDUAL and SPECIFIC ASPECTS or issues regarding design. The broader and more comprehensive the list of aspects to which we relate our design method, the more complete will be our description and the more thorough our knowledge and awareness of our view of design.

I. Just as we hold certain issues or aspects of people or buildings as being particularly important to KNOWING or DESCRIBING them, we also probably hold particular aspects about design as being of more importance than others. The identification of what we consider to be these CRITICAL ISSUES is a key goal in expressing our view of design.

IV. PROGRAMMING – DESIGN MODEL

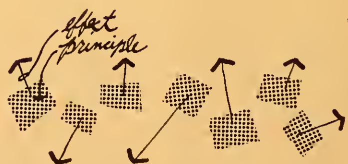
A. This text was written with a view of design in mind. The model essentially involves the identification of a series of RELATED and SEQUENTIALLY DEPENDENT events which lead to an architectural product. As programming is PART of this sequence, the event chain provides a context for defining the ROLE of programming in PLANNING.



B. The view of design sequence used is as follows;

1. Reality (laws, principles).
2. Search for and discovery of laws and principles (fact-making).
3. Known facts.
4. Gathering of facts.
5. Analysis, evaluation and organization of facts into meaningful patterns.
6. Response to facts in design synthesis.
7. Building product.
8. Building consequences.
9. Evaluation.

C. REALITY. Both research and programming assume the existence of objective reality. They depend upon the fact that there are laws and principles which govern cause-effect relationships and that these laws exist independently of our awareness of them.

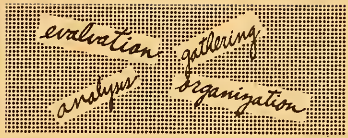


D. RESEARCH. It is the objective of research to uncover these laws to allow us to predict and control the consequences of our design decisions.

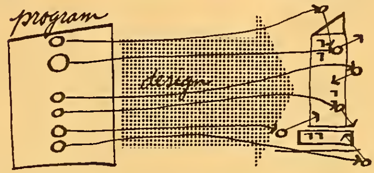
E. FACTS. Out of research, facts are "produced." Although we are never absolutely certain of them, still they provide us with a basis for making choices with some assurance of

the outcome. There are many categories of facts. They range from natural or physical laws (those governing structural design), to "man made" facts (traffic laws).

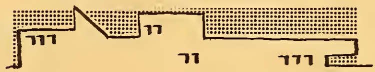
F. GATHERING, ANALYSIS, EVALUATION AND ORGANIZATION OF FACTS. These form the core of programming in architecture. They are essentially concerned with insuring that as many of the important consequences of the building design as possible are anticipated and planned for so that the building is successful in these critical respects.



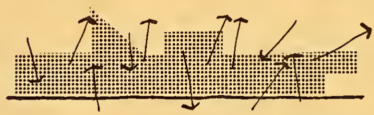
G. RESPONSE TO FACTS IN DESIGN. The planning of the building is based upon the establishment of the desired building effects or consequences in programming and the creation of the physical product which will most effectively bring about those consequences. The more comprehensive the designer's program the more knowledgeably he can plan his product.



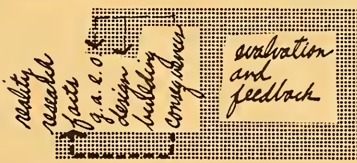
H. BUILDING. The physical product of the design process is not the designer's final concern. The consequences of the building are in the last analysis the critical issue in design.

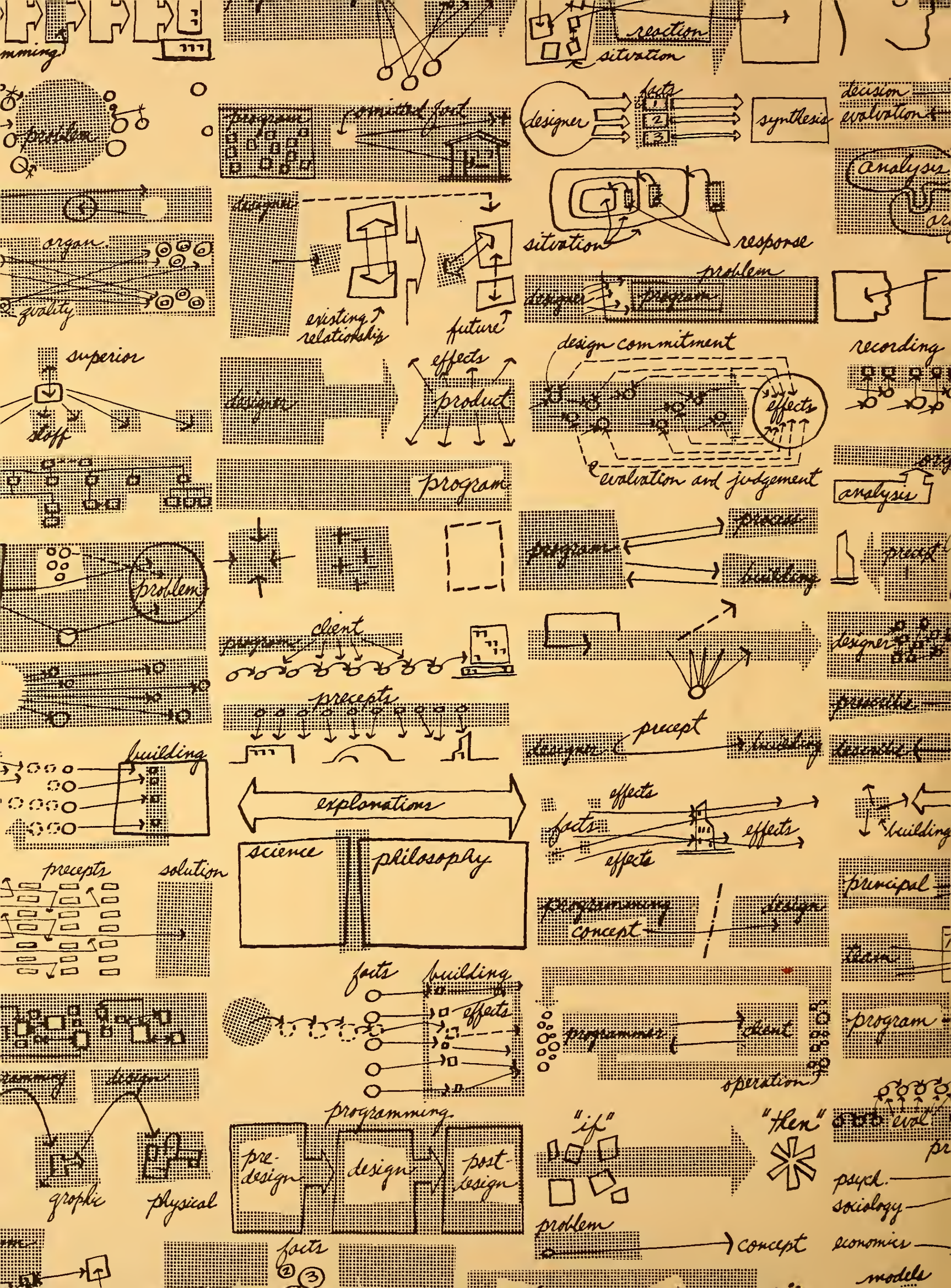


I. BUILDING CONSEQUENCES. Buildings will have their effects whether planned for or not. Because a fact has not been considered in programming or design will not prohibit it from having its consequences.



J. EVALUATION. This is an effective method for expanding our awareness of consequences of individual design decisions and building features. In effect, evaluation is a form of research and serves as a feedback mechanism to facts, programming and design. Evaluation and feedback loops also occur between every event in the sequence.




reaction
situation
problem
program
committed fact
designer
facts
1
2
3
synthesis
decision
evaluation
analysis
organ
quality
designer
existing relationship
future
situation
response
problem
designer
program
superior
staff
designer
effects
product
design commitment
effects
evaluation and judgement
recording
program
analysis
process
program
building
problem
client
program
designer
precepts
precept
designer
building
explanations
science
philosophy
building
effects
facts
effects
effects
building
precepts
solution
programming
concept
design
principal
facts
building
effects
programmer
client
operation
program
design
programming
pre-design
design
post-design
"if"
"then"
eval
graphic
physical
problem
concept
psych.
sociology
economics
models
facts
②
③

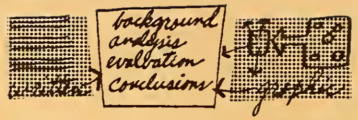
10

BACKGROUND

SURVEY OF PROGRAMMING

I. DEFINITION

A. A program usually takes the form of a WRITTEN AND GRAPHIC document wherein background information, fact analysis and evaluation and conclusions pertinent to a project are organized and presented.



B. The specific CONTENT AND FORMAT of a program may change depending on the nature of the project.

C. No matter what form it takes or project it addresses, the INTENT of a program is always the same.

D. A program is a PLAN OF ACTION for defining and achieving desired results and goals (consequences).

E. The program INTENT may be a new building, orderly operations or facilities growth, improved operational efficiency, better working environment or informed choice of site location.

F. Although the TRADITIONAL programming type is that which is prepared for the design of a new facility, programs may also take several OTHER forms.

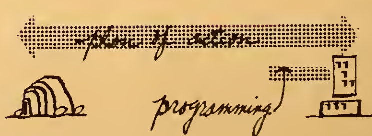
1. A LONG RANGE PLAN assesses present conditions, projects current trends and outlines future potentials regarding a client's operation and building development.

2. A FEASIBILITY STUDY may involve issues such as timing, phasing or advantages and disadvantages regarding site selection and acquisition or building expansion versus remodeling.

3. OPERATIONS ANALYSIS can be applied to overall efficiency, cost-benefit issues, staffing projections, alteration or expansion of services, equipment purchases, quality control or environmental inventories.

4. A PROGRAM for a new building serves as a tool for recording client needs, insuring that these needs are met and evaluating the building design before construction begins.

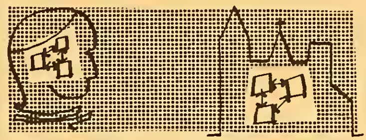
G. In its broadest definition as a "plan of action," programming has existed for as long as architecture itself. Programming in its present roles and forms, however, is a relatively RECENT development. There are several possible reasons



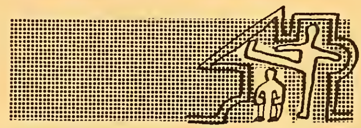
why programming has lagged in its maturation as a discipline.

1. "Primitive" building largely dealt with immediate and personal needs (shelter). The needs were those of the builder, and programming, design and construction occurred virtually SIMULTANEOUSLY.

2. Even as construction techniques became more sophisticated, they still housed relatively simple functions with which the designer-builder was often very FAMILIAR (religious structures). There was no need to write down what he already knew about what was to be housed in the building.



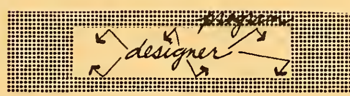
3. As building tasks became more complex, they were often subjugated to the "formal" qualities of the building. The functions were a reason to "make a work of art." The designer's knowledge of what was to happen inside could be SUPERFICIAL.



4. The view of a building as a "whole" kept the NUMBER and COMPLEXITY of individual concerns in design relatively small. This also delayed the need to be systematic in documenting the many variables involved.

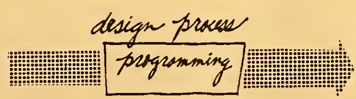
5. Allied fields such as psychology and sociology had not developed to the point where they could add to the list of building CONSEQUENCES which the designer must be aware of. With relatively few effects to concern himself with, a program wasn't really necessary.

6. Architects have regarded the architectural program as RESTRICTIVE. Many see no direct correlation between the program document and their own operations in the design process.



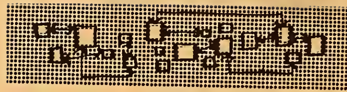
7. Programming has not been considered as a DISTINCT architectural service in terms of FEE STRUCTURE. Many firms cannot afford to do a comprehensive programming job.

H. Although there are still many improvements to be made, programming is recognized today as an ESSENTIAL part of the planning process for most design situations. This is largely due to several factors.



1. Architects are now faced with the task of designing buildings which must house FUNCTIONS about which they know little or nothing.

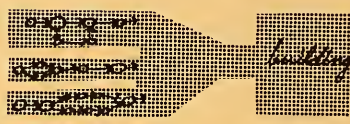
2. There is an increased need for MULTI-FUNCTIONING buildings whose operations are extremely complex and whose variables defy an unsystematic approach to planning.



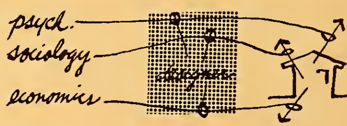
3. The architect is required to take responsibility for more and more DETAILED planning in his projects. The number of individual decisions to be made is becoming increasingly unwieldly.

4. Much more is being demanded of buildings in terms of PERFORMANCE. An "exciting piece of sculpture" or "pleasant composition" is no longer sufficient justification for the design, construction and maintenance costs incurred by the client. Programming is an important step in insuring that the building "performs."

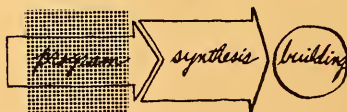
5. The growing view of buildings as a SYNTHESIS OF SUBSYSTEMS has resulted in the identification of many "parts" of the "whole" which can be studied, evaluated and designed for. This has provided programming with SUBJECT MATTER.



6. The rapid advances made in ALLIED FIELDS which have established many facts in terms of man-environment relationships have greatly increased the scope of building CONSEQUENCES which the designer must take into account.

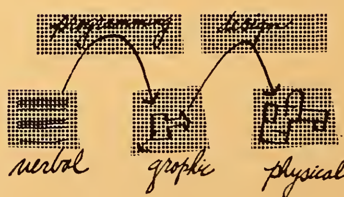


7. There is increasing recognition that the design process for solving a problem is directly linked with the PROBLEM STATEMENT and that the key to a succesful building lies as much in having a good PROGRAM as in good design SYNTHESIS.



II. PROGRAMMING ROLES

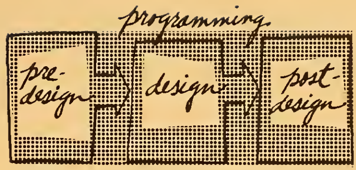
A. The most critical ROLE of programming is the purpose it serves in the "view of design" system outlined in the Introduction. This role will be discussed more fully in later sections. Briefly, in terms of the design paradigm mentioned, programming finds, selects and organizes pertinent facts and translates them from VERBAL to GRAPHIC expression so that they may, in turn, be translated into a physical expression.



Programming is a vital segment of the chain of events leading to the PREDICTION and attempted REALIZATION of valued building CONSEQUENCES.

B. One convenient way of organizing the roles of program-

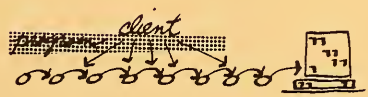
ming is in terms of their TEMPORAL relationship to the act of planning a building. Generally programming roles may be PRE-DESIGN, DESIGN and POST-DESIGN. There are many simultaneous roles that a program may play. Widely differing roles become mutually exclusive or detrimental when the program becomes specially tailored for very unique purposes.



1. Pre-design

   PRIOR to the start of the building design process, a program may:

   a. Serve as a PROMOTIONAL package for the client.
   b. Be used to promote client staff MORALE.
   c. Function as a CATALYST for discussions before governmental approval boards.
   d. Serve as a COMMUNICATIVE TOOL between the client and the design firm.
   e. Define the client's NEEDS in terms that can be translated into design issues during building planning.
   f. Provide the basis for PRESENTATIONS to interested civic groups.
   g. Help to organize the DECISION-MAKING responsibilities of the client related to building planning.
   h. DOCUMENT the client's project budget, organizational and operational structure and record recommended improvements.
   i. Provide the client with a FRAMEWORK for outlining his future needs and requirements.
   j. Serve the design firm as a framework for UNDERSTANDING the client's operation.
   k. EDUCATE the client regarding the planning process and provide him with an understanding of the reasons behind design decisions to be made.
   l. Avoid OVERESTIMATION of furniture, equipment and space needs.



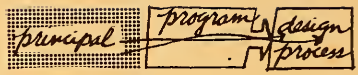
2. Design

   DURING the planning process a program may:

   a. Direct the building PLANNING PROCESS.
   b. Aid in generating viable ALTERNATIVE building designs.
   c. Serve as a vehicle for active CLIENT PARTICIPATION in the planning process.
   d. Help insure a GOOD FIT between client operations and the building.
   e. Determine building QUALITY and SCOPE based on BUDGET and TIME limitations.
   f. Promote a THOROUGH PLANNING RESPONSE to

the needs of the client, especially in projects of large scope or great complexity.

g. Function as an EVALUATIVE tool for investigating and testing different planning approaches.
h. Give the designer an INSIGHT into the "spirit" of the problem.
i. Serve as a catalyst in fostering a CREATIVE approach to the problem.
j. Provide a basis for RESOLUTION of differences with the client during planning.
k. Function as a mechanism for design CONTROL for architectural principals who are not actively involved in the planning process.



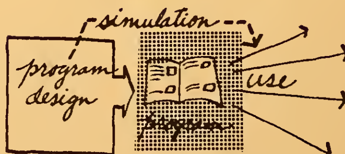
3. Postdesign

AFTER the design process is complete, a program may:

a. Provide the client with a TOOL FOR EVALUATING the design proposal.
b. Insure the most ECONOMICAL building design within the problem requirements.
c. Result in a facility planned for GROWTH AND CHANGE.
d. Serve as a manual for the USE AND OPERATION of the new facility.
e. Allow the client to ORGANIZE and DIRECT his future rather than merely reacting to situations and needs as they occur.
f. Insure maximum operational EFFICIENCY and PRODUCTIVITY for client functions in the new facility.
g. Maximize the opportunity for the new building to contribute to its URBAN and ECOLOGICAL surroundings.

C. One role not mentioned above is as a PROMOTIONAL and EDUCATIONAL tool for the programming or design firm.

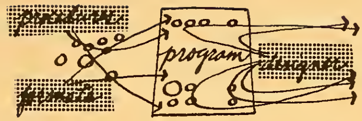
D. A program may or may not be put into PUBLISHABLE form depending on its PURPOSES. Simulation of the use of the document in its respective roles is vital in designing the program. Oftentimes the very same data will appear in different FORMS and FORMATS because of its use for different TASKS.



III. PROCESS

A. The process of programming is composed basically of GATHERING, ANALYZING, EVALUATING, ORGANIZING and PRESENTING information pertinent to the design problem.

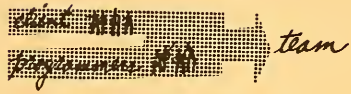
B. The PROCEDURES and FORMATS in programming are intended to organize and outline the factors relevant to predicted desired BUILDING CONSEQUENCES and to present these factors in a way that the designer may easily UNDERSTAND and USE.



C. The PROGRAMMING firm need not be the DESIGN firm.

D. The specific operations performed in programming will depend upon the program TYPE and PURPOSE.

E. In building facilities programming the programming TEAM is composed of representatives of the client and programming firm.

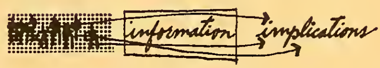


1. To insure effective programming and expedite the process, team members must have AUTHORITY to make decisions.

2. The client group is responsible for providing information about their operational NEEDS.

3. The programming firm is responsible for GATHERING, ANALYZING, EVALUATING and ORGANIZING pertinent information.

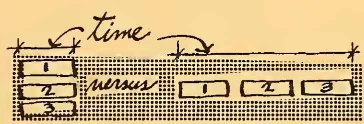
4. Together, the team members review the functional and organizational IMPLICATIONS of the information.



5. The team approach facilitates the evolution and testing of INNOVATIVE changes in the client's operations.

6. The team approach insures that the program will be a JOINT EFFORT of the client and programming firm.

F. Many work tasks within the programming process are carried out SIMULTANEOUSLY rather than sequentially to shorten total programming time.



G. In addition to various other introductory information, the program format basically includes GOALS, FACTS, PRECEPTS and CONCEPTS.

1. GOALS include the purpose of the project, client and user goals and the project description.

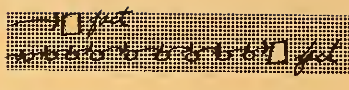
2. The FACTS involve both quantitative and qualitative information and issues.

3. QUANTITATIVE data may encompass site, climate, codes, utilities, zoning, project scheduling, space re-

quirements and building quality and scope in relation to budget.

4. QUALITATIVE information may pertain to activity analysis, sensory considerations and desired environmental qualities.

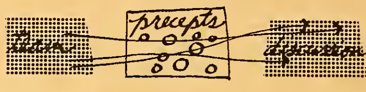
5. Some facts are IMMEDIATELY available (description of client operation) while others must be DERIVED (space needs).



6. PRECEPTS are individual planning commitments dealing with important quantitative and qualitiative factual issues.

7. The precepts serve as criteria for EVALUATING design alternatives and ELIMINATING those not in sympathy with the initial programmatic and design ASSUMPTIONS.

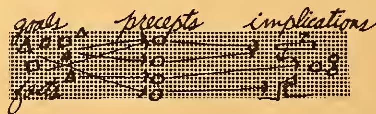
8. The precepts are generated by the programming team members and provide a method for discussing and arriving at DECISIONS about critical project issues.



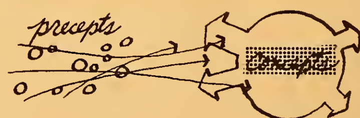
9. Some precepts are reasonable and logical on FACE VALUE while others demand considerable STUDY before accepting them as design assumptions.

10. Precepts inevitably contain VALUE JUDGMENTS made by the programming firm based on the "spirit of the problem" and other difficult-to-document factors.

11. Precepts are the direction-giving part of the program and suggest to the designers what the ARCHITECTURAL IMPLICATIONS of the goals and facts might be. They are in essence mini-design commitments.

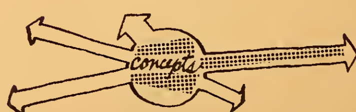


12. Taken together, the precepts are meant to suggest overall planning directions or CONCEPTS. The concepts suggested may be a LITERAL extension of the precepts or an INTERPRETIVE one.



13. CONCEPTS are general planning directions suggested by the goals, facts and precepts.

14. There may be SEVERAL viable concepts possible that answer the critical issues and precepts. The program should clearly indicate which seems the MOST VALID.



15. At the conclusion of the development of ALTERNATIVE concepts, and the recommendation that ONE of these be selected, the program is complete.

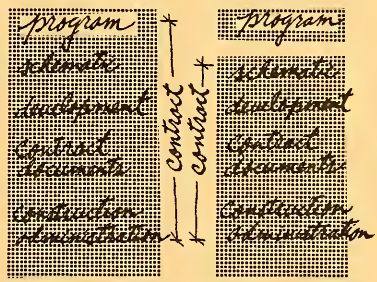
16. The responsibility for the FURTHER development of a concept into a BUILDING DESIGN is SEPARATE from the programming responsibility.

H. A good program should include more than an accumulation of NEUTRAL facts and actually extend into the realm of DESIGN commitments and recommendations.

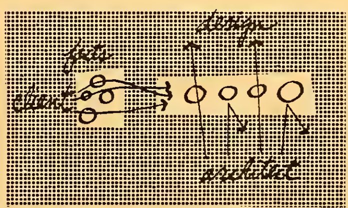
IV. PROFESSIONAL ASPECTS

A. The definition of programming as an ARCHITECTURAL SERVICE and the description of how the architect should be COMPENSATED for this task is still unclear in the profession. This reflects the general lack of agreement in the profession regarding what constitutes a "basic" programming service and what constitutes an "additional" service. (Basic services are performed as part of the SCHEMATIC DESIGN fee. Additional services warrant EXTRA compensation).

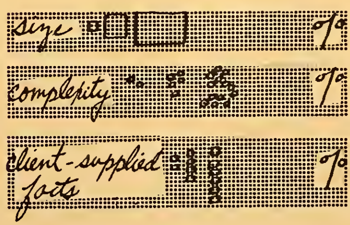
B. Some architects believe that all programming services should be performed as part of their responsibility to design and build the BEST building possible. For them, there is NO additional compensation for programming. Others feel that the increased complexity of buildings and the growing amount of details which an architect must design for make it unreasonable to assume that the ever increasing programming time should be ABSORBED into the basic fee. These architects often write a SEPARATE CONTRACT for the programming phase of the job with compensation for this work being in ADDITION to the basic fee for design.



C. Clients are often able to supply much of the needed programming information themselves. Some are capable of executing almost all of the programming work pertaining to their operation. Generally, however, a large percentage of programs done by clients are NOT of value to the architect. The success of a client-executed program depends largely on the client's knowledge of his organization and his ability to state his needs in terms which are MEANINGFUL to the architect.



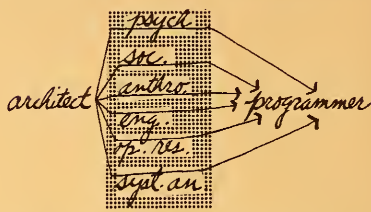
D. In a limited survey, it was found that the average cost of programming to the client is between ¼ and ½ percent of the construction cost of the building. This, of course, may vary with the SIZE of the job, the COMPLEXITY of the needs and the amount of data that the CLIENT can supply. Programming firms vary in the manner in which they contract to do their work. Methods include a percentage of the estimated construction cost, cost plus expenses, and predetermined total amount.



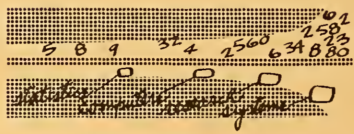
E. Because programming is demanding greater and greater sophistication in terms of gathering and organization techniques, there is an increasing number of firms that SPECIALIZE in programming. Many of these limit their work to SPECIFIC building types (hospitals, schools) while others are more general and diverse in their work.

F. Usually, only the LARGER architectural firms are able to offer comprehensive programming services to clients representing complex organizations.

G. The qualifications of a programmer vary from firm to firm. Some feel he should be an ARCHITECT because he must communicate with DESIGNERS. Others feel he should NOT be an architect because he will be biased in his programming. Several programming firms use psychologists, sociologists, anthropologists, engineers, operations researchers, and systems analysts. It can be assumed that a combination of architecture with any of these would be advantageous.



H. As programming is becoming a more QUANTITATIVE discipline, it would be beneficial for a prospective programmer to have as much exposure as possible to statistics, computer science, principles of basic research and systems analysis.



I. Some of the people who are currently very active in developing architectural programming as a DISCIPLINE are:

1. William Pena — Caudill, Rowlett and Scott, Architects
2. W. R. Matthews — Matthews and Associates, Architects
3. Lester Gorsline — Lester Gorsline and Associates, Programmers
4. Gerald Davis — The Environmental Analysis Group
5. Edward Agostini — Becker and Becker, Planning Consultants
6. Christopher Alexander — Center for Environmental Structure
7. E. Todd Wheeler — Perkins and Will, Architects
8. C. Herbert Wheeler — Pennsylvania State University
9. Ben H. Evans — Building Research Institute

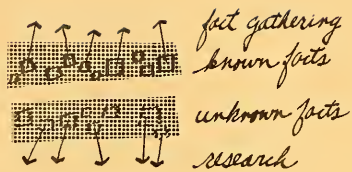
RESEARCH

I. DISTINCTIONS

A. Research: careful, systematic, patient study and investigation in some field of knowledge undertaken to establish FACTS or PRINCIPLES.

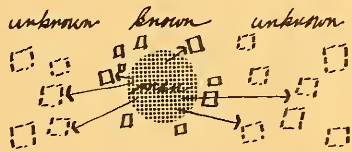
B. Research may be BASIC (above definition), or APPLIED, APPLIED research attempts to take facts uncovered by BASIC research and find useful applications for them.

C. Research is distinct from data or fact gathering (as done for example in programming) in that the latter involves accumulating and organizing facts which are KNOWN, while the goal of the former is the discovery of NEW facts. The one is attempting to make a CONTRIBUTION to knowledge while the other is making use of EXISTING knowledge.



II. ASSUMPTIONS, VALUES AND ATTITUDES

A. Research assumes the existence of facts (laws and principles) as INDEPENDENT of our awareness of them. Man is "immersed" in these laws and is governed by them in that they determine the consequences of actions. Man does not "make" basic natural laws but is faced with the task of finding out what they are.

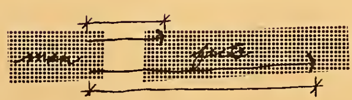


B. The "facts" discovered by research are never absolute certainties. They are at best, statements of PROBABILITIES for certain effects, given certain situations.

C. We value research because by isolating and identifying cause-effect relationships we are better able to CONTROL and PREDICT those effects which we value and depend upon.

D. Research generally is QUANTITATIVE in nature. Relationships can be stated more exactly in this way. Probabilities can be expressed more precisely with the use of numbers. Research in some fields lends itself to mathematical models more readily than others. Many researchers feel that this is because the qualitatively-oriented fields have not developed far enough to be able to use the mathematical mode.

E. The invention and refinement of techniques which allow us to EXTEND our senses are vital to the continued success of research (microscope, telescope, spacecraft).



F. There are some general VALUES and ATTITUDES characteristic of researchers as a group:

1. flexible in their beliefs
2. tentative in their conclusions
3. beliefs based on evidence and not authority
4. value knowledge of underlying reasons for phenomena
5. skeptical
6. tolerant
7. value honesty and accuracy in reporting data
8. detached emotionally as much as possible from their work
9. individualists
10. dedicated
11. value knowledge as an end in itself

There are many intrinsic PLEASURES of research which relate to the maturation of the scientist.

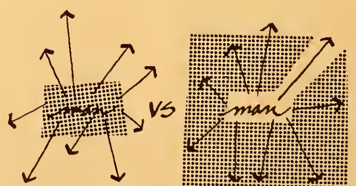
1. curiosity
2. delights of ambiguity and uncertainty
3. contest with nature
4. escape from boredom of everyday experience
5. aesthetic pleasure
6. joy of exercising the intellect

STATUS in the research community is dependent upon several factors:

1. stage of development of the discipline
2. role played in research (theorist ranks higher than scientist, basic research higher than applied research)
3. originality and influence on others (including impact of contributions to the field)
4. institution with which the scientist is associated

G. Scientific concepts have no INTRINSIC moral content.

H. An important issue for many researchers to resolve for themselves is that of FREE WILL vs. DETERMINISM as it relates to research. Since human action is part of the world's phenomena, should it not be included as one of the objects of control and predictability? Does free will and unfettered choice diminish with the growth of data about the human mind?



fatalist: "Certain results are destined to happen no matter what a person does."

determinist: "There are functional relations between variables and this knowledge can be used to predict the future (to predict the consequences of design

decisons in architecture). There must be some degree of determinism for an individual to have free will, since he has to be able to choose from among predictable behaviors."

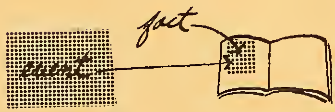
I. "Scientific laws are not PRESCRIPTIVE — that is, they don't say HOW people OUGHT to act. Scientific laws are DESCRIPTIVE. They DESCRIBE how people and things DO ACT."

III. RULES

A. In research, no DECISIONS are made on the basis of faith, power, monetary rewards or self-protection.

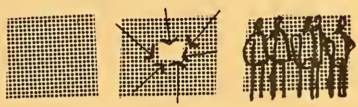
B. Science is distinguished from dogma in that science isbased on FACT. Dogma is based on BELIEF.

C. A "fact" is the actual occurrence of an EVENT. It is singular and happens at a given time. After it occurs it is gone forever. "Data" is the recording of that fact in some SYMBOLIC form.

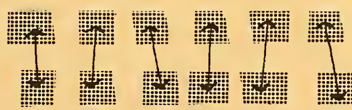


D. Criteria for accepting events as FACTS:

1. Must be singular.
2. Available to public scrutiny.
3. Different individuals can know what the event was that is being described.



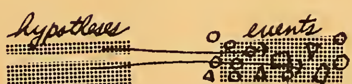
E. Scientific laws are descriptions of relatively constant RELATIONSHIPS between certain kinds of phenomena. Laws are established by the consistent repetition of relations between kinds of events and not by a singular occurrence of any succession of events.



CRITERIA for accepting a statement as a scientific law:

1. Must be about kinds of events and not directly about any singular event.
2. Must be a large amount of data supporting the law and little or none discounting it.
3. Must show a functional relation between two or more kinds of events.
4. Relation should be applicable to very different events.

F. An important goal in research is to develop the SMALLEST set of hypotheses or principles which will account for the GREATEST variety of events.



G. For any law in science we find that at some level it rests on INDUCTION. An ASSUMPTION is made that since

the event has occurred before on several occasions, under SIMILAR conditions it will happen again. Regularity does not guarantee certainty, and all induction is based on regularity.

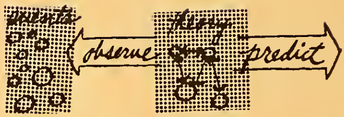
IV. METHODOLOGY

A. Sequence

1. casual observation
2. identification of area of concern
3. suspicion of cause-effect relationships not previously uncovered through experiment
4. formulation of hypothesis or tentative theory
5. testing of hypothesis
6. hypothesis disproved or accepted as a theory

B. Remarks

1. A THEORY is the basic formal system developed to account for observation. The purpose of a theory is to describe and explain observable events and to PREDICT what will be observed under certain specified conditions.

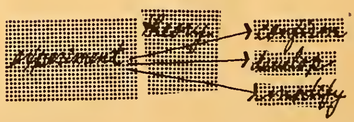


2. A tentative theory is needed in research to provide the scientist with a FRAMEWORK for experimentation.

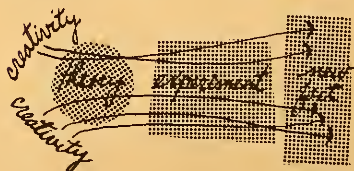
3. Experimental design consists of three factors:

   a. independent variables — directly manipulated by the experimenter to effect dependent variables
   b. dependent variables — measures taken during the experimental process
   c. control variables — should not vary systematically from condition to condition (constants)

4. Experimental results provide data that is used to confirm, develop or modify a theory. A theory tends to be confirmed if actual observations agree with those PREDICTED by the theory.



5. Theories about phenomena we see play an important role in directing the research of scientists. "Concept-getting" is at the heart of research progress and points to the need for CREATIVITY in science. Before experimentation can begin, an hypothesis must first have been formulated. This is the point at which "discoveries" begin.

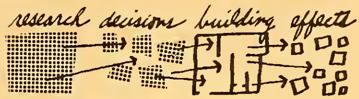


V. ARCHITECTURAL RESEARCH

A. The rules and methodology of research in general apply

to ARCHITECTURAL research as well.

B. The development of research in architecture is largely due to the broadening of architectural involvement into fields more advanced as scientific disciplines. These provide the subject matter and rigor needed for research.



C. Architectural research is intended to establish greater certainty about the CONSEQUENCES of specific design decisions so that those decisions may be made more knowledgeably and with greater predictability.

D. The list of topics for possible doctoral research at universities is a good indication of those areas in architecture which are felt to have enough "substance" for research. Representative categories are:

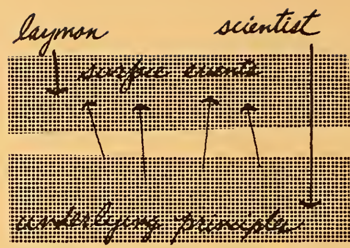
1. Architectural Design and Design Process
2. History and Philosophy of Architecture
3. Building Technology
4. Behavioral Science
5. Urban Design
6. Facilities Design (specialty in a specific building type)
7. Architectural Operations
8. Man-Environment Relationships

PHILOSOPHY AND FACTS

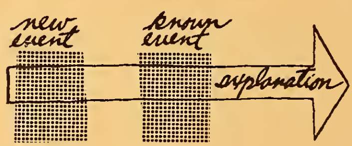
I. DISTINCTIONS

A. Fact: "the state of things as they actually ARE: reality, actuality, truth."

B. In defining facts as "the way things are," we can distinguish between existing or past DESIRABLE conditions and UNDERLYING principles or laws that brought them about. What is seen on the "surface" is a RESULT of cause-effect relationships. For example, to a layman, facts are what is experienced and perceived. To the scientist facts are those deeper principles removed from the level of what we perceive which actually GOVERN what we perceive. The scientist is involved in underlying, causative relationships.

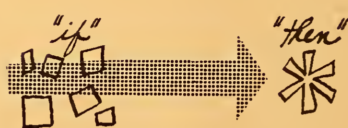


C. Another viewpoint defines facts as METAPHORS. This definition sees "facts" as expedient means for explaining and categorizing perceived new phenomena in terms of KNOWN phenomena. Facts here have no relationship to any "reality" or "truth" but serve simply as a system of REFERENTS.



The facts we know are composed of both METAPHORIC and actual CAUSE-EFFECT relationships.

D. Facts, as the term is used here will always connote cause-effect relationships. These relationships can be expressed as, "IF a given situation, THEN a resulting effect." Basic laws and principles are not altered by our failure to discover or understand them.



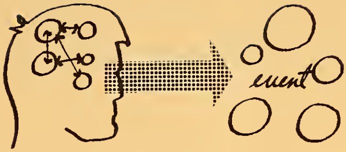
E. The belief that there exists an objective reality independent of our awareness of it is an ASSUMPTION. It cannot be proven with absolute certainty. The assumption is based on the fact that we can identify repetition in the effects of certain actions, but repetition does not assure that given the same situation the same effect will result. All choice and action are based on a predicted outcome and so DEPEND upon the assumption of an independent reality.

II. PHILOSOPHY AND FACTS

A. Philosophy: "theory or investigation of the principles or laws that regulate the universe and UNDERLIE all reality." Philosophy deals with hypotheses which attempt to explain why things are the way they are, not in terms of

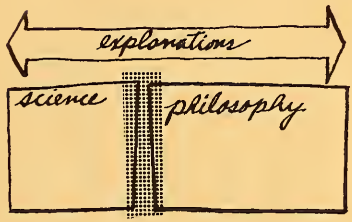
empirical research but by constructing broad explanatory frameworks based on logic and reason.

B. Man seems to need to EXPLAIN the causes of those things he values and depends upon. This form of control or possession of that which is valued is often evident in the prevalent philosophies of different periods in history. The TYPES of things that are explained and the NATURE of the causes proposed provide insight into the values of a culture.

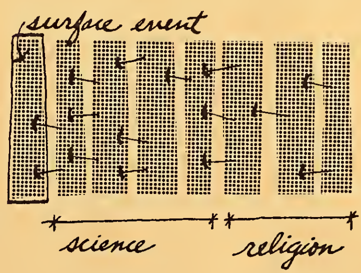


C. Earliest recorded philosophy deals with pleasure, pain, good and evil and the laws and principles which must be followed to achieve what is valued. No matter what the goal of a philosophy, it always sets forth a SYSTEM of cause-effect, action-consequence relationships, a system for explaining the "nature of reality."

D. In proposing explanations of events, philosophy usually begins just beyond the frontier of scientific discovery. At any given point in time, research is able to trace cause-effect relationships only so far. The INTERFACE between philosophy and science is at this frontier. Philosophy proposes the way things are beyond where science is able to empirically probe. As science widens and deepens its domain, philosophical assumptions are proven true or otherwise.



If explanations (facts) are thought of as existing on different LEVELS ranging from immediate causes to deeper, more removed causes, this provides a way of describing the so-called "conflict" between religion and science. Because it has claimed explanation of causes "near the surface" of observed events, religion has appeared to have retreated as science has advanced. This has not meant that religion is invalid, only that it has underestimated the NUMBER of levels of discoverable causation behind events before the concept of "first-cause" can be discussed.

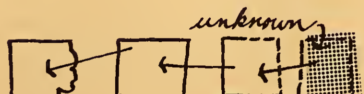


III. LEVEL OF FACTS

A. Depending upon our viewpoint, there are different levels at which facts exist. Each level has to do with more BASIC cause-effect relationships as facts become more REMOVED from "surface events."

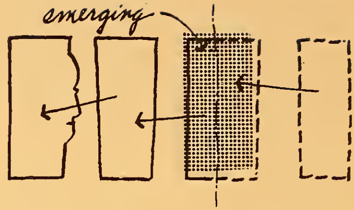
One method for outlining these levels follows:

1. Unknown facts — laws and principles which are as yet UNDISCOVERED (aspects of brain chemistry, molecular structure, astronomy and physics). The devel-

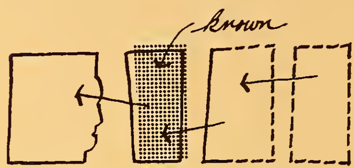


opment of our ability to extend our senses will be largely instrumental in new discoveries.

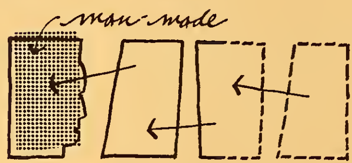
2. Emerging facts — principles which are in the PROCESS of being tested for their validity. If these prove to be EXPLANATIONS, they will become known and usable facts. We can then use these as a basis for making decisions with some assurance of PREDICTABLE outcomes. Emerging facts represent the furthest that science has been able to probe into the causes of surface events.



3. Known facts — these are all the unchanging or "natural" relationships that we have been able to discover. They serve as a basis for DECISIONS. There are many "levels" of known facts due to the receding nature of cause-effect relationships. For any surface event there is a chain of events which led to and caused it. Each "link" in the chain is a fact.

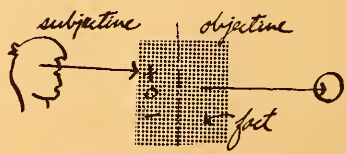


4. Man-made facts — the levels of facts up to and including "known facts" have all pertained to relationships which have no dependence upon our awareness of them. "Man-made facts" are our REACTION to them. Man-made facts are principles or laws that we institute to regulate our behavior with respect to known facts (building codes, structural formulas and traffic laws).

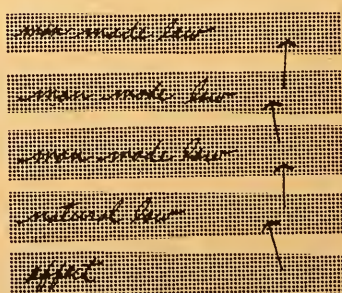


Man-made laws or facts must be based upon a CORRECT assessment of the consequences of "known facts" if they are to produce DESIRED results.

The effects of known facts are neutral. We make man-made facts based on a VALUE JUDGMENT about these effects. (The causes of physical pain are "natural laws." Whether pain is considered a positive or negative experience may depend upon the cultural situation).

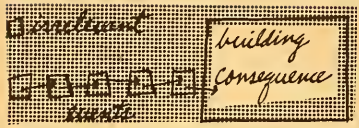


In a complex society, the man-made facts that are based on the consequences of natural laws sometimes become so far REMOVED from their original intent that it becomes difficult to find their real meaning. Man-made laws become "layered" where new laws are instituted based on existing man-made laws which can ultimately be traced to the effects. Values begin to rest upon these removed concerns as they used to rest on the ACTUAL natural consequences. New needs are created which are in essence ARTIFICIAL. We begin to deal with the symbols of the consequences as though they were the consequences themselves (suicide at bankruptcy).

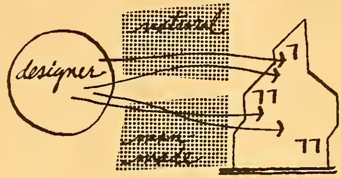


IV. FACTS IN ARCHITECTURE

A. In gathering the facts which relate to a given design problem, a key issue is that of RELEVANCE. A fact is only relevant to programming and design if it is part of a chain of events or cause-effect relationships that lead to an effect or consequence that we judge as important.

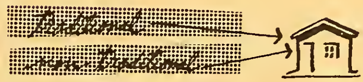


B. In architectural design we are faced with both NATURAL and MAN-MADE facts. All may be viewed as "if . . . then" situations. As a designer becomes more aware of the consequences or effects of his design decisions he enables himself to make his decisions more knowledgeably and confidently and to more easily achieve the result that he predetermines as desirable.

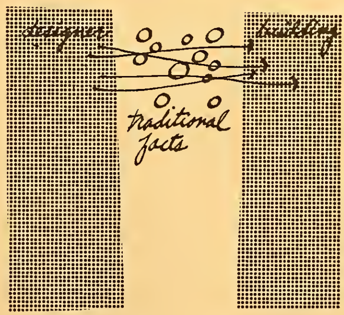


C. The relationship between design DECISIONS and building CONSEQUENCES is vital to architectural programming and design. Programming serves to gather the facts, evaluate their relevance to the situation, identify the effects they may have on each other and organize them for the designer's use in design synthesis. Design synthesis attempts to make a physical product whose consequences are those called for in the program.

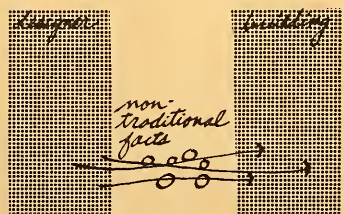
D. The facts pertinent to an architectural design situation can be classified as TRADITIONAL and NON-TRADITIONAL.



Traditional facts are those which we customarily include on our list of concerns when programming or designing. These may include activity patterns, people involved, furniture and equipment needed, site information, climate information and perhaps desired effects of the building form and environment on its inhabitants. The effects of our decisions with respect to SOME of these facts we feel confident about. For others we may know what we want the consequences to be but are not sure of the WAY to produce them. This is especially true in matters that involve psychological reactions to the environment we create. It may even be true for some of the areas that we consider familiar (functional efficiency).

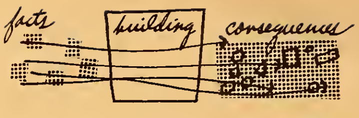


Non-traditional architectural facts are those that are PERTINENT to design (they involve building consequences) but not ordinarily considered in programming or synthesis. The growth of non-traditional architectural facts is largely due to research in ALLIED FIELDS such as psychology, sociology, anthropology and physics. They may involve relationships such as light level-work efficiency, desk orientation — psychological security or glass additives — glare reduction.



These may seem too "detailed" for the architectural designer to concern himself with. Nevertheless, the decisions he makes in programming and design RESULT in an environment that has these types of CONSEQUENCES.

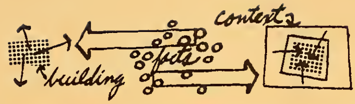
If it is of value to know what the EFFECTS of our designs are, then it is important to become more familiar with non-traditional architectural facts.



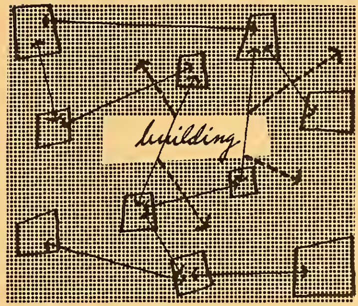
NON-TRADITIONAL FACTS

I. GENERAL CONSIDERATIONS

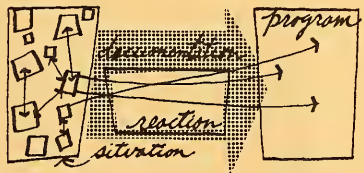
A. For any given building there is a spectrum of facts that are relevant to the CONSEQUENCES that the building will have and that its context will have on the building.



B. The building can be thought of as an ADDITION to an existing set of cause-effect relationships (existing car and pedestrian traffic on and around the site, site drainage to adjacent property, existing foilage, sunlight, noise, scale, image, activity tempo, client functions, activity patterns of clients' workers such as driving to work or going to lunch). The building becomes PART of many of these situations. For some, there is little change, while others are altered drastically. The addition of a building to these situations can be likened to a relative coming to live with a family permanently. It is important to know how the "addition" may ALTER the existing systems or patterns of events.

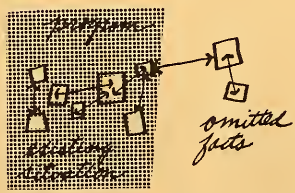


C. One of the functions of programming is to document the "existing situation" in its broadest sense. The program should also include some REACTION to the different "existing situations" in terms of the value of preserving or altering them. This is of great help to the designer in setting his objectives (determining what building effects would be desirable).

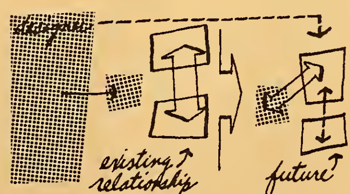


D. Generally, facts have a twofold importance in programming and design:

1. The omission of a fact in programming about the "existing situation," whether due to negligence, inexperience or because the relationship has not been discovered (is not a known fact), may result in building consequences that are both UNANTICIPATED and UNDESIRABLE. (Unhappy design accidents usually far outweigh the happy ones when designing from incomplete data).



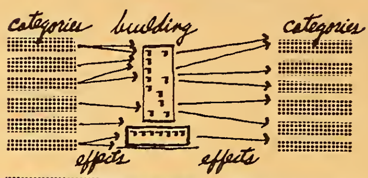
2. Assuming the rare situation where the designer is fully aware of all the networks of relationships in the existing situation, if he is to AFFECT those relationships as intended or desired, he must also have a knowledge of the CONSEQUENCES of specific design decisions about the physical building (effect of scale on existing area image, effect of space on psychological reaction of workers, effect of layout on client function effi-



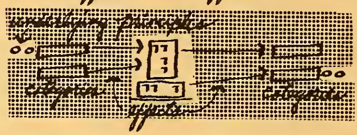
ciency or effect of materials combinations on visual unity).

The need for this knowledge also applies to the effects ON the building by the existing situation (climate on materials, activities on maintenance, snow load on structure or sunlight on thermal comfort).

E. Although the number and types of "building on situation" and "situation on building" effects are many, the general CATEGORIES of these effects are fairly traditional (function, site, climate, form, light, materials, structure, openings, mechanical). Within each of these groups we are aware of many individual cause-effect relationships or "facts."



Some of these we assume as "rules of thumb" without really knowing the UNDERLYING principles involved. For others we are aware of the principles to a certain depth beyond the surface event.

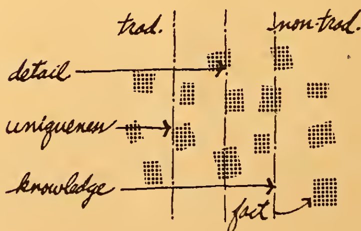


F. In discussing the broad spectrum of facts which are pertinent to building design, it is sometimes helpful to make a distinction between "traditional" and "non-traditional" facts.

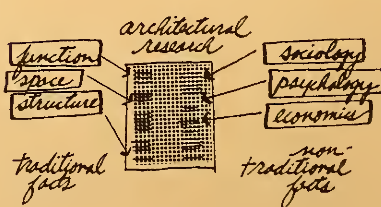
Traditional facts are those that we CUSTOMARILY include on our list of concerns when programming and designing. Non-traditional architectural facts are those that are relevant to design (they involve building consequences), but are NOT ordinarily considered in programming and synthesis.

II. NON-TRADITIONAL FACTS

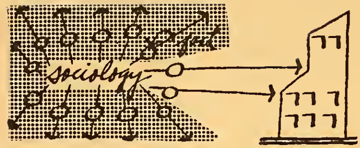
A. There is no clearcut division that can be made between traditional and nontraditional architectural facts. The classification of a fact as one or the other will depend upon the degree of programming and design DETAIL required for the building type in question, the UNIQUENESS of the building type and the depth and breadth of the KNOWLEDGE of the designer. What is non-traditional for one building or designer may be very common for another.



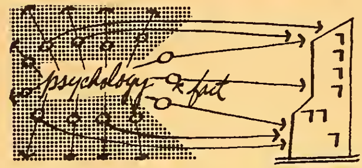
B. Research in fields ALLIED to architecture is primarily responsible for the discovery of non-traditional architectural facts (psychology, sociology, anthropology, physics, engineering, systems engineering, business management, finance, economics, computer technology, industrial processing). The discovery of cause-effect relationships under the title of "architectural research" has occurred largely in these fields.



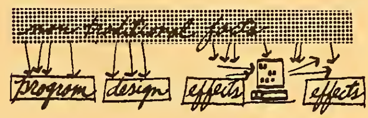
C. When dealing with facts generated by another field, the issue of RELEVANCE becomes important. There is a temptation to try to apply the whole field to architecture even though much of it may not be pertinent. It is important to SCREEN facts from related fields in terms of their relevance to building consequences.



D. Because these related fields are seldom concerned with APPLYING their findings to architecture, it is equally important not to overlook facts because their architectural implications aren't immediately evident. The continued generation of non-traditional architectural facts largely depends on our sensitivity to these sometimes HIDDEN or REMOTE relationships.



E. Non-traditional architectural facts may be applicable not only to the effects BY and ON a building but also to the process of PROGRAMMING and DESIGNING it (systems analysis, computers, decision theory).



III. AREAS OF CONCERN

A. With respect to the "levels" at which facts exist, non-traditional facts are unknown, emerging, known and man-made.

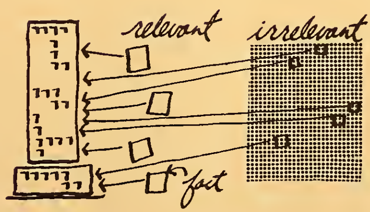
B. Related to the traditional architectural concerns (function, site, climate) non-traditional facts include:

1. whole new CATEGORIES of cause-effect relationships (radiation protection system for moon structures)

2. new developments within TRADITIONAL areas of concern (plastics, adhesives, office landscaping)

3. remote levels of UNDERLYING laws or principles of "rules of thumb" within traditional fact categories (molecular causes of paint deterioration)

4. minute or subtle building CONSEQUENCES which the programmer or designer seldom is able to concern himself with. Though they may in fact have an impact on the effects of the building, there are many facts which have so little to do with the important building consequences that they warrant no consideration. Taken alone they may be relevant. The judgment of the programmer or designer may render them irrelevant.



C. Areas where research is discovering RELATIONSHIPS applicable to architecture include man-environment, building materials, techniques of assembly, economics and design

process. Example relationships that might be classified as non-traditional to architecture are:

1. role of physical environment in learning
2. effects of visual order versus complexity on learning
3. effects of centralization vs. decentralization of workers on efficiency
4. effect of worker group size on performance
5. relationship between topic of conversation and conversion distance
6. effect of background brightness on visual acuity
7. effect of specific colors on visual comfort
8. effect of visual clutter on visual efficiency
9. relationship between sound frequency of speech and intelligibility at receiver
10. noise frequency-sonic annoyance relationships
11. relationship between continuous and random noise and performance
12. effects of random noise on boredom
13. relationship between exterior image and customer buying patterns
14. relationships between natural land features and settlement patterns of high income families
15. effects of government involvement in housing on construction techniques
16. effects of new shopping centers on surrounding areas
17. effects of new CBD construction on the municipal budget
18. physical effects of sunlight on architectural surfaces
19. actual effects of large amounts of glass at exterior walls on mechanical equipment costs
20. effects of fire on architectural materials

21. effects of washing machines on individual sewerage disposal systems

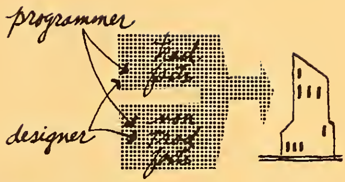
22. effects of exterior plastics on interior thermal comfort

23. effects of new adhesives on traditional architectural materials

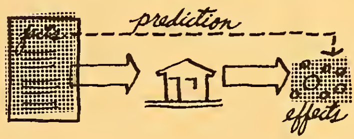
24. relationship between the use of mathematical models in design and building programming

25. effects of new decision theory on sequence of concerns in design synthesis

D. Given the recognition that non-traditional facts are relevant to building design, it behooves the programmer and designer to expand and deepen their awareness of them insofar as possible. Ideally, by staying abreast of current developments these facts will become SECOND NATURE, much as our traditional facts have largely become. It is important to at least be familiar with SOURCES that may be used for specific projects as the need arises.



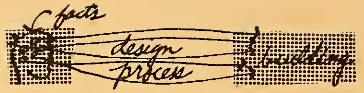
E. Ideally, there should be NO non-traditional architectural facts. They should be as FAMILIAR to the designer as the traditional ones so that the effects of our buildings can be controlled and predicted more accurately and comprehensively.



TRADITIONAL FACTS

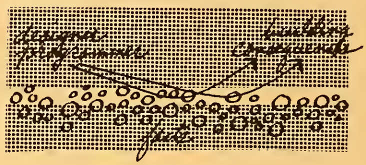
I. GENERAL CONSIDERATIONS

A. Traditional architectural facts are those that we "usually" CONSCIOUSLY deal with in programming and designing a building.

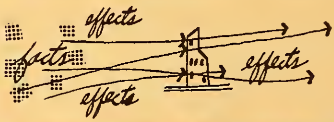


B. The requirement of a designer to be involved with more than the traditional architectural facts is largely dependent upon the degree of DETAIL required in planning and the UNIQUENESS of the project.

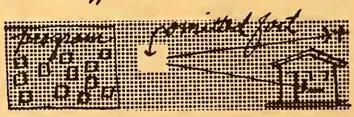
The more that is required of a building in terms of PERFORMANCE and the more important it is to be ACCURATE in predicting the building consequences, the less adequate traditional architectural facts become. This is to say that the "usual" involvement of the programmer and designer in building consequences is relatively SUPERFICIAL.



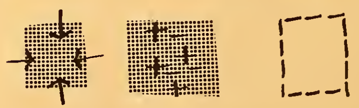
C. Like any facts, traditional architectural facts determine the effects of the building on the existing situation and vice versa. They are important in directing and controlling BUILDING CONSEQUENCES.



D. Failure to consider a fact may result not only in NEGATIVE consequences on or by the building, but also in some potential POSITIVE consequence not being brought to fruition.



E. Traditional facts may be descriptions of the EXISTING situation, an EVALUATIVE statement about the existing situation (preserve or alter) or a statement as to desirable FUTURE situations or consequences.

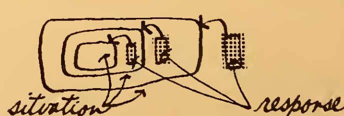


When a statement serves as a RULE for making design decisions and for evaluating those decisions after synthesis, it is called a PRECEPT.

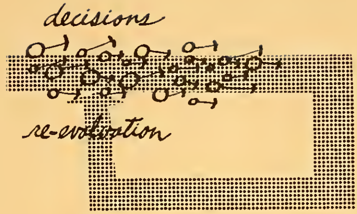
Precept: A rule or maxim to DIRECT actions or decisions.

In programming a precept is a directive for the DESIGNER to strive to achieve some building consequences or situation.

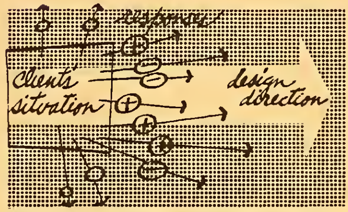
F. It is sometimes a convenient model in organizing our design experience to think of the synthesis process as a progressive "response to the existing situation." It begins in programming with the documentation of the "situation" as brought to the ARCHITECT by the CLIENT. Through his evalua-



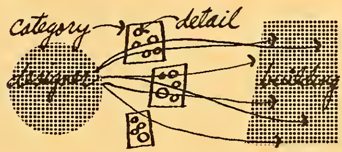
tive reaction to the client's situation, the PROGRAMMER adds to the "existing situation" that which the DESIGNER must respond to. The designer's first conceptual responses to the program expand the "existing situation" even further. As design decisons are made they become the existing situation or "facts" to which subsequent responses must be made. Feedback and evaluation loops allow us to UNDO the "existing situation" to different degrees and begin the process anew when needed.



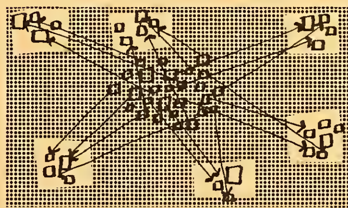
G. It seems clear then that the evaluative responses of the programmer to the client's existing situation are highly instrumental in setting the DIRECTIONS of design synthesis. In the same way, the early stages of synthesis become DETERMINANTS for those decisions which come later. Even with recycling and feedback, the early stages of design are critical. The first "view of the problem" is the beginning of the CONVERGENCE process leading to the chosen solution.



H. The more comprehensively aware of not only GENERAL categories but SPECIFIC details in traditional facts, the more thorough and efficient the designer can be. He can also avoid the frustration of having to REDESIGN conceptually because of a more detailed "fact" that he simply wasn't aware of.

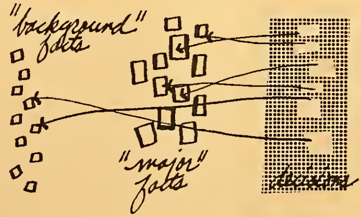


I. Individual facts don't intrinsically belong to any "family" or category. Depending upon which of their QUALITIES is important in a situation, we group them differently. The choice of how to group facts in programming is an EVALUATIVE design act in itself. It reflects "how we see the problem" and is a prelude to how we will go about solving it. Information GROUPINGS may be a more important determinant in the conceptual stages of synthesis than the individual facts themselves.

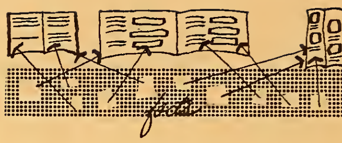


J. Specific facts may be more pertinent to conceptualization than to design development or vice versa. Some facts are PRIME ORGANIZERS, while others are SECONDARY.

K. "Background facts" which form the governing context of the design situation (client goals) often prove important in making specific design decisions. These are especially useful where there seems to be no immediate criterion for making a decision. These types of facts which at first may seem remote from the "front line" of synthesis decisions may often be the only BASIS for making important judgments about very specific building issues.



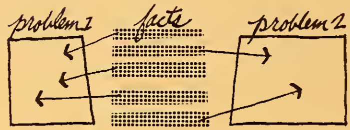
II. TRADITIONAL FACTS



A. Different facts may be pertinent to different types of PROGRAMMING DOCUMENTS. In the same way that we screen facts in terms of their RELEVANCE to building consequences, we also evaluate their PERTINENCE to the purpose of the document where they will be contained.

Some of the different types of programming documents in architecture are:

1. master plan
2. long range plan
3. site feasibility
4. building program
5. comprehensive plan
6. project definition.



B. Below are some TYPICAL traditional architectural fact categories. For any specific situation some are more relevant than others. Groupings may also be different depending on the problem (pertain to and involve important building consequences).

1. Similar projects and critical issues.

   a. past projects of similar function, circumstance and scope
   b. critical issues involved in the building type
   c. trends in the field

2. Client

   a. client goals
   b. philosophy of the organization
   c. goals of the client's process — sub-goals to achieve main goals — user goals
   d. staff organization and framework — personnel diagram
   e. rank and role of personnel
   f. major departmental divisions within the organization —role of each—goals and sub-goals within the overall process
   g. critical issues involved in the organization (people to people relationships, "channels")
   h. does organization actually operate the way it is structured?
   i. divergence of present operations from expressed goals — possible improvements
   j. degree of achievement of sub-goals
   k. individuals or committees responsible for planning with architect—role and responsibility in decision-making

l. related (non-client) organizations which might affect planning
m. impact of change or growth of related organization

3. Financial

a. budget — firmness, degree of flexibility
b. funding methods — bonds, loans, fund raising
c. timing — construction costs, escalation, interest rates, concurrent similar projects taxing public support
d. construction phasing—prices, local construction market, strong and weak local trades, incremental construction
e. design requirements of lending institutions
f. comparative cost data on similar projects which have been constructed

4. Building Codes

a. occupancy allowed
b. structural loads allowed
c. exits required
d. stairs (number, type, access, fire rating, size, minimum distances to reach stairs)
e. fire ratings required of materials
f. ventilation — openings
g. toilets (number and fixtures of each)
h. fire sprinklers
i. alarm systems

5. Planning by related organizations

a. duplication of services
b. review boards
c. approval boards (regulations, by-laws, planning criteria)
d. projected construction of similar projects

6. Function

a. operational systems—including links beyond the building
b. critical issues in insuring success in systems' operation
c. needs which are supporting to operation (lounge, waiting, toilet, janitor)
d. main operational sequences — "feeder sequences" which support main sequence
e. divisions or departments in the system
f. general departmental relationship affinities
g. number and type of people involved (task categories)
h. operations performed by each type of person
i. systems of people movement

(1) points of origin and destination
(2) frequency and pattern (continual or intermittent)
(3) degree of urgency
(4) role in the overall operation
(5) peak loads

j. systems of information movement

(1) points of origin and destination
(2) frequency and pattern (continual or intermittent)
(3) degree of urgency (speed required)
(4) role in the overall operation
(5) form
(6) storage implications
(7) operations performed on information (including production and removal of trash)
(8) peak loads

k. systems of material movement

(1) points of origin and destination (including delivery and pickup)
(2) frequency and pattern (continual or intermittent)
(3) degree of urgency
(4) role in the overall operation
(5) form (size, weight)
(6) special considerations (fragile)
(7) operations performed on material (including unpacking and disposal of waste)
(8) storage implications
(9) peak loads

l. work nodes (stations where work is performed)

(1) number, type and relationships
(2) number and type of people at each
(3) nature of tasks performed

(a) key issues in successful performance of tasks
(b) identification of possible sources of strain in performing tasks

(4) furniture and equipment required for each person (including visitors, clients)
(5) accessories required for each person
(6) sizes, electrical requirements and other considerations regarding furniture, equipment or accessories

(7) area requirements of each node
(8) circulation patterns within each node (people, material, information)
(9) security requirements (open, closed, locked)
(10) general electrical requirements at each node
(11) criteria for selecting architectural surfaces and detailing
(12) special relationships with other work nodes (visual control)
(13) lighting requirements

(a) intensity required at task
(b) incandescent vs. fluorescent
(c) direct sun vs. indirect
(d) skylight vs. window
(e) need for total darkness
(f) need for controlled lighting

(14) sensory

(a) type and intensity of stimuli produced (noise, odors, vibration, dust, electro-magnetism, bacteria)
(b) type and intensity of stimuli which must be excluded or screened (including visual privacy)
(c) important environmental situations (mood, atmosphere)

(15) air conditioning requirements

(a) heat generated by equipment and people
(b) special air circulation or ventilation requirements (isolation, 100% exhaust, decontamination)
(c) special temperature requirements
(d) air additives
(e) special controls over air conditioning
(f) grouping of similar air conditioning requirements
(g) total needs
(h) space required for mechanical
(i) vibration control
(j) heating and cooling seasons

7. Site

a. legal description of property (boundaries, dimensions, rights of way, deed restrictions, easements, curbs, curb cuts, hydrants, poles)

b. zoning

(1) present allowable uses
(2) setbacks
(3) access points
(4) relation to street lights and median breaks
(5) density
(6) heights allowed
(7) parking required

c. utilities

(1) locations
(2) distances to site
(3) depths
(4) telephone, gas, water, sewer, electrical
(5) capacities (present and projected)

d. soil conditions

(1) percolation
(2) bearing
(3) chemicals
(4) density

e. land contours

(1) elevations
(2) drainage patterns (including from and to adjacent land)
(3) flood basins (tides)
(4) blocked visual access due to mounds and ridges
(5) points of visual emphasis
(6) flat areas
(7) slope orientation to surrounding areas (visually)

f. significant features

(1) rock outcroppings
(2) existing buildings
(3) ditches
(4) water
(5) trees

g. existing foliage

(1) tree types
(2) limb spread
(3) height
(4) ground cover (where drainage may be affected)

h. sensory

(1) noise (direction, intensity, frequency, pattern, probability of continuance)
(2) odors (direction, intensity, pattern, type, probability for continuance)
(3) visual (poor views, good views, public and private zones, reliability of continuance of view)

i. time-distance

(1) car - pedestrian
(2) to and from significant points on and around site
(3) time-distance on site

j. existing pedestrian traffic on and around site

(1) volume
(2) location
(3) frequency and pattern (time of day, continual, intermittent)
(4) nature (to work, school, lunch, random stroll)
(5) possible contribution to these activities

k. existing vehicular traffic on and around the site

(1) volume
(2) location
(3) frequency and pattern
(4) nature
(5) possible contributions to these activities

l. surrounding physical environment

(1) surrounding zoning
(2) possible development on adjacent and surrounding property
(3) profile (skyline)
(4) scale
(5) image
(6) materials
(7) forms
(8) density
(9) light (shade and shadow)
(10) orientation (views of site from other points)
(11) landscaping forms
(12) details
(13) geometry (existing paving patterns, building edges and heights, axes, walls, modules and rhythms)

m. surrounding social environment

(1) identifiable patterns
(2) ethnic groups and values
(3) relationships between groups

n. shadow patterns on the site (trees, adjacent buildings)

o. parking and site circulation

(1) needs (present and projected)
(2) area required
(3) dropoffs required at entry
(4) lighting
(5) special controls (restricted parking)
(6) on-site circulation required (between buildings)
(7) supporting circulation (to lunch, to work)
(8) volume and frequency patterns (peak loads)
(9) patterns of direction of entry approach and departure (people and cars)
(10) existing roads
(11) points of logical access-egress (all types of traffic)
(12) surrounding land values

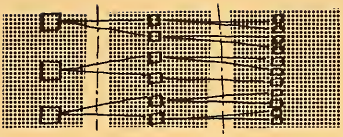
8. Climate

a. rainfall (frequency, volume, patterns)
b. sunlight (critical vertical and horizontal angles)
c. temperature (seasons, extremes)
d. wind, breezes (seasons, directions, velocity, extremes)
e. snow (seasons, volume, patterns)
f. humidity (seasons, percentages)
g. potential natural catastrophes (tornado, hurricane, earthquake, flood)

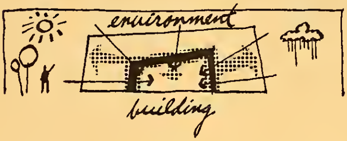
9. Growth and Change

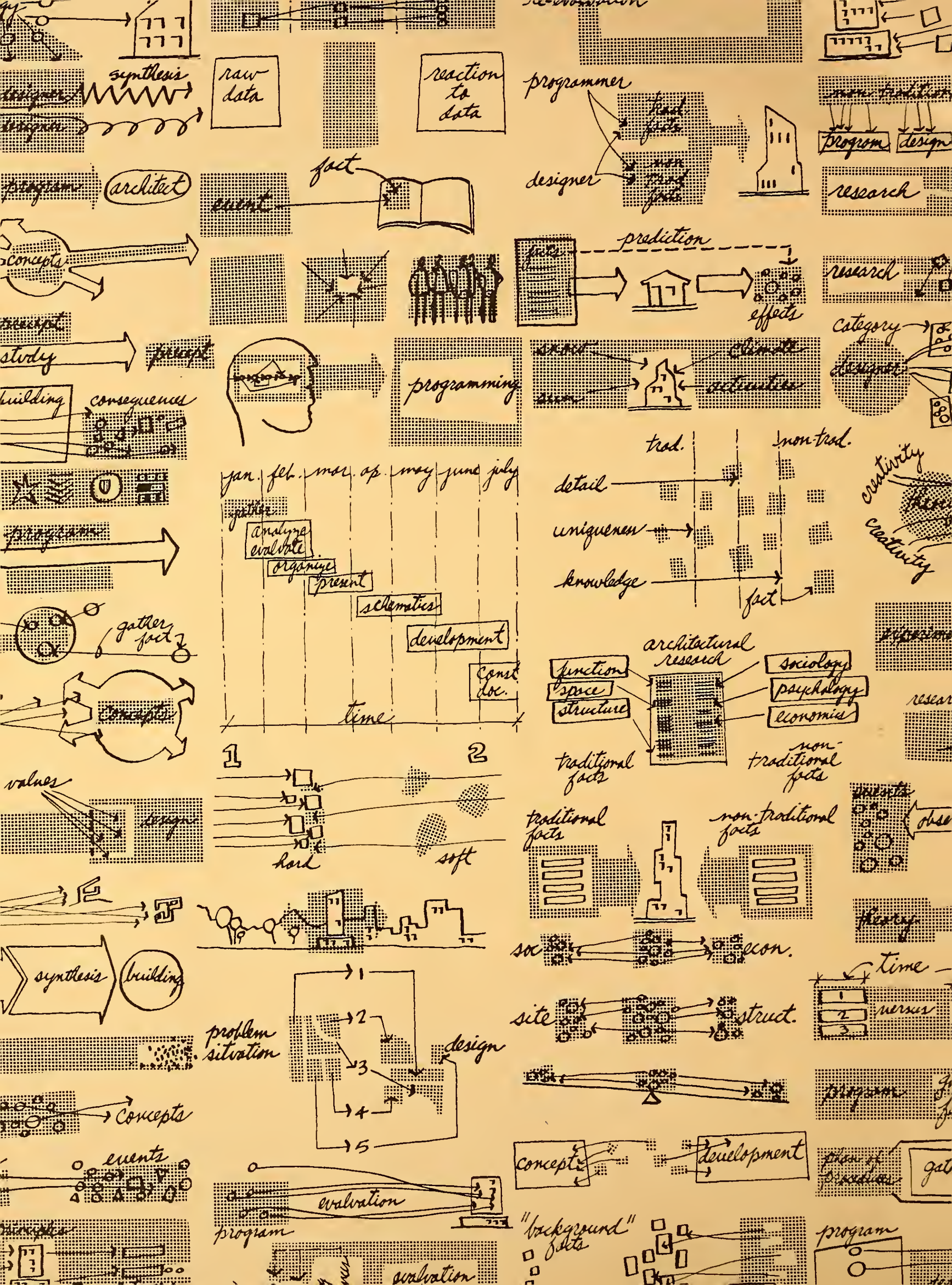
a. present and projected supporting market or public served
b. projected staffing (number and type)
c. projected goals and supporting sub-goals
d. anticipated deletion of departments and addition of new departments
e. areas of expected changes in operations (layout and building perimeter implications)
f. projected changes in information or materials systems (disposables)
g. influence of growth and change of one department on all others
h. future area needs (construction, cost, design and parking implications)
i. projected utility needs — comparison with present and projected supply capacities

C. Each of these fact categories may be EXPANDED to more DETAIL depending on the design requirements. There are also many other fact categories not listed here that pertain to some of the other programming FORMS (long range plan).



Every fact category and specific fact contained under its heading involves CONSEQUENCES which the building has on its environment and contained functions and which the environment has upon the building.




synthesis
designer
raw data
reaction to data
programmer
hard facts
non hard facts
designer
program
design
research
architect
fact
event
concepts
prediction
facts
effects
research
category
designer
concept
study
precept
programming
climate
activities
building
consequences
trad.
non-trad.
creativity
creativity
jan. feb. mar. ap. may june july
gather
analyze evaluate
organize
present
schematics
development
const. doc.
time
detail
uniqueness
knowledge
fact
program
gather fact
concepts
architectural research
function
space
structure
sociology
psychology
economics
traditional facts
non-traditional facts
values
design
1
2
hard
soft
traditional facts
non-traditional facts
research
soc.
econ.
synthesis
building
time
site
struct.
versus
1
2
3
problem situation
design
1
2
3
4
5
concepts
program
events
concept
development
evaluation
program
"background" facts
program
evaluation

INFORMATION GATHERING

ANALYSIS, EVALUATION AND ORGANIZATION OF FACTS

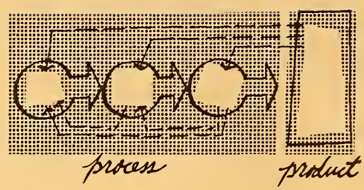
DESIGNING FROM THE PROGRAM

PROGRAM AND DESIGN EVALUATION

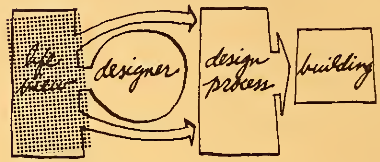
INFORMATION GATHERING

I. CONTEXT

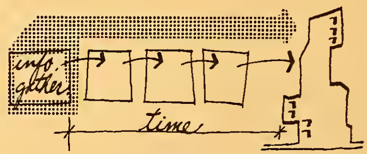
A. The quality of a PRODUCT is determined by the quality of the PROCESS that produced it. A building is the result of operations performed in the design process. Its actual limitations and achievements are "prescribed" before construction begins. If thought of as simply one end of a series of actions and decisions performed through time, we can see the value of not only studying buildings as PRODUCTS but also the OPERATIONS that make them.



B. The specific operations performed in programming and design that finally describe the future physical product to be built are limited or influenced by the BROADER views held by the designer. His framework for ordering his EXPERIENCE IN GENERAL has implications on his models for IDENTIFYING and MANIPULATING the elements of design.



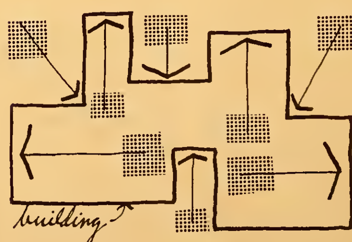
C. Information gathering is the start of the "formal" programming process. Although possibly remote from the final design in time, it has a very real effect upon the character of the resulting building. Included here are some of the values, operations and relationships involved in "gathering" as a link in the chain of design events that prescribe the CONSEQUENCES on and by the resulting building.



II. GENERAL CONSIDERATIONS

A. In relation to our design model FACTS can be thought of as "consequence categories." They are the areas of concern wherein the building AFFECTS and is AFFECTED BY what surrounds it and what it contains.

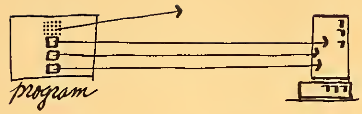
B. The gathering of facts in programming assumes there are EXISTING DATA which must be allowed to be influential in making the building design. The degree to which the designer ALLOWS the facts to form the building will depend on his design philosophy. In the same way, the programmer may FORM his gathering format and collected facts to a greater or lesser degree depending on his attitudes about his role in the design process ("let the problem SOLVE ITSELF" versus "it is the function of the programmer and designer to GIVE the problem order").



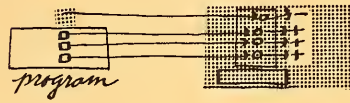
C. Although the particular approach or model for gathering information is essentially a product of the programmer's

DESIGN VIEW, there are some qualities about this operation that seem to generally be desirable.

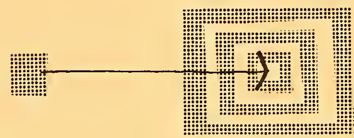
1. Relevance - Facts gathered should be PERTINENT to the CONSEQUENCES on or by the building. Irrelevant data causes inefficiency and confusion in gathering, analysis, design and evaluation.



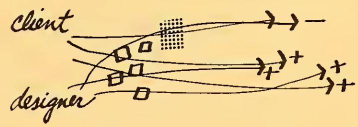
2. Completeness - It is important to have ALL the pertinent data at hand when designing. An incomplete program can result in design omissions and erroneous conclusions regarding the required BUILDING TASKS.



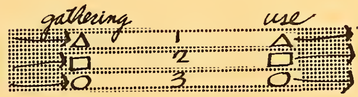
3. Accuracy - This quality is especially important when there are surveys or other statistical studies that will be used later in making other design FACTS (precepts, conclusions). It also applies to the recording of information from all sources including qualitative statements by the client and users.



4. Clarity - Clarity is vital to insuring good communication with the client about the facts as we see them. This also relates to giving the designer a CLEAR statement of determinants that both he and the client UNDERSTAND and AGREE UPON.



5. Usability - The gathering sequence and the forms used for recording data should relate to when and how it will be used in programming analysis, organization, and design synthesis.



6. Efficiency - Wasted motion, materials and time and retracing of steps should be MINIMAL.

D. In discussing data gathering as a programming operation, it is convenient to divide it into FOUR general groups of concerns.

1. planning of procedures.
2. outlining of data to be collected.
3. design of forms and formats.
4. definition of sources and execution.

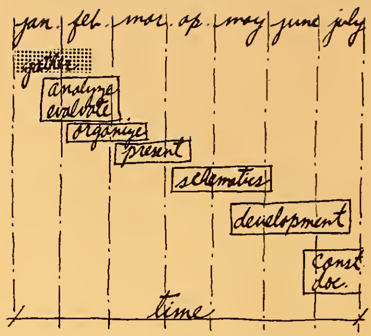
III. PLANNING OF PROCEDURES

A. This operation is sometimes called "defining the program for the program." It is the design of HOW we plan to go about gathering our information.

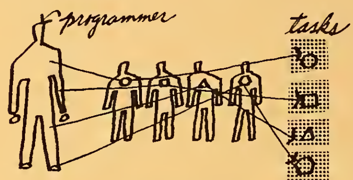
B. As in all design operations the planning of procedures is largely dependent upon the DESIGN VIEW of the program-

mer. There are, however, some concerns that can apply to data gathering in general.

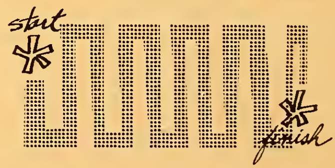
1. A plan of procedure for gathering information must relate to the overall TIME FRAMEWORK for the job. Information analysis, organization and presentation, schematics, design development, construction documents and construction, all come after and depend upon this first step. They all have their time allotments based on the overall job organization and budget. The success of the project for the architect as well as the client largely depends on execution of all the design steps within their ASSIGNED time frame. Planning of data gathering cannot be separated from the planning of the WHOLE project. Intermediate dates for the completion of different gathering tasks and the use of critical path planning may be helpful.



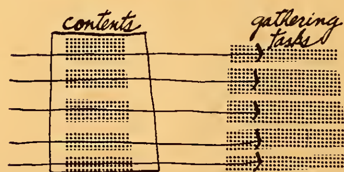
2. Before a plan of procedure can be undertaken, the programmer should first know HOW MANY people will be assisting him and what their QUALIFICATIONS are for certain tasks. A complex project requiring many "gatherers" creates yet another need: that of organizing the communication between the STAFF during the gathering process.



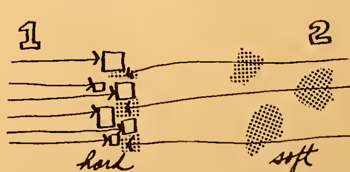
3. A plan of procedure deals with what must be DONE. It should be stated in terms that describe OPERATIONS. This is absolutely essential where gathering is to be done by several people. Questions such as "where do you start?" and "how do you know you're finished?" must be answered by the plan of procedure.



4. It is sometimes helpful to begin a plan of procedure by projecting or anticipating what the CONTENTS and FORMAT of the final document will be and working backwards to methods for getting the needed information. The definition of a detailed TABLE OF CONTENTS for the program is usually an excellent way to organize gathering tasks.

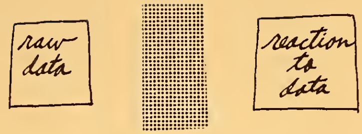


5. In any data gathering situation there are some facts that are FIXED and others that are TENTATIVE. In the interest of efficiency it may be helpful to gather fixed or "hard data" first. This type of information often provides the basis for "firming up" the tentative facts and usually constitutes many of the critical determinants in DESIGN.



This issue relates to the distinction that can be made between RAW data or facts and facts that are CONCLUSIONS or REACTIONS to the raw data (precepts, evaluative statements) which result in secondary or once-

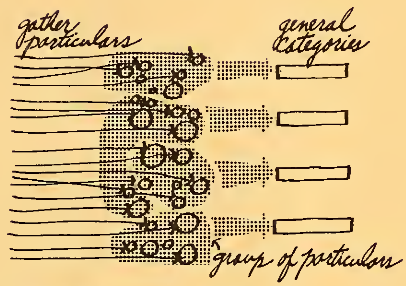
removed information. The programmer must be careful to distinguish in his document between what is FIRST HAND raw information and what is, in effect, his OPINION about or REACTION to information.



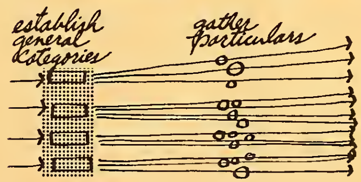
C. The use of MODELS or "concepts of approach" for gathering information is one of the clearest illustrations of how a view of design affects specific operations. Four issues that relate to the formation of a model for data gathering are:

1. particulars to generalities versus generalities to particulars.
2. segregated gathering versus integrated gathering.
3. immediate fact evaluation versus deferred fact evaluation.
4. atomistic synthesis versus wholistic synthesis.

D. The PARTICULARS TO GENERALITIES approach gathers individual and specific facts and makes no groupings or larger categories until after all the pertinent "particulars" are gathered. The assumption here is that "generalities" are composed of "specifics." Generalities have no meaning except as TITLES for particulars that possess similar qualities. Individual facts must be known before broad conceptual frameworks can be constructed. To artificially set the categories ahead of time would jeopardize possible linkages between facts that have ARBITRARILY been put in different categories.

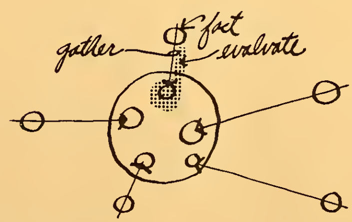


In GENERALITIES TO PARTICULARS the assumption is made that since we will eventually STRUCTURE the facts that we should be able to establish these broad categories ahead of time. This point of view assumes that the programmer is an active "form giver" to the information and that the giving of that form may occur at any level of facts, general or particular.

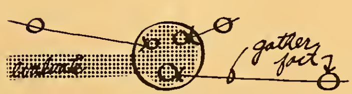


E. Facts may be evaluated AS they are gathered (immediate) or AFTER the gathering process is complete (deferred).

1. In IMMEDIATE EVALUATION, facts are studied for linkages, relationships, and hierarchies and groupings are made "as we go." Values are placed on the data and precepts are formed based on the facts the programmer has AT THAT POINT in his gathering progress. This approach assumes that in any design problem there are facts which are "prime organizers" for synthesis and that the sooner these are identified, the sooner the synthesis process can begin.



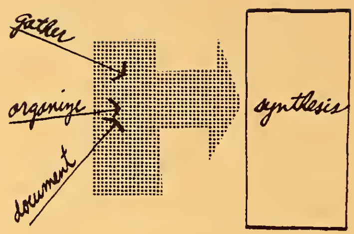
2. DEFERRED EVALUATION involves putting off the grouping, sorting, hierarchy linkages and relationships until "all the facts are in." It assumes that it is of value to check for relationships between facts on all levels in



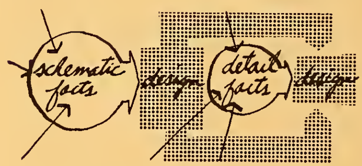
all categories and to form values and precepts based on a knowledge of the WHOLE PICTURE. Prime organizers are not uncovered here until gathering is essentially complete. This viewpoint is tempered by the recognition that we NEVER can be absolutely certain when all the facts are in.

F. Fact gathering may be either SEGREGATED or INTEGRATED with design synthesis.

1. SEGREGATED GATHERING requires a comprehensive gathering, organization and documentation of facts BEFORE design synthesis. It is based on the assumption that even the first design decision should not be made without ALL the facts. To do so is to risk the possibility of having to undo design decisions because some "detail" doesn't come to light until well into the design synthesis process. This attitude argues that it is unreasonable, for example, to document space needs without knowing what is needed in the spaces.

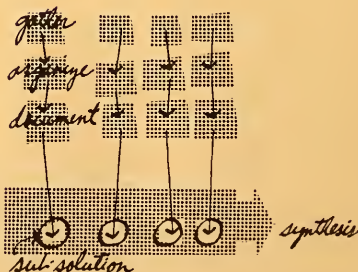


2. INTEGRATED GATHERING assumes that conceptual design decisions require only "overview" data and that specific information need not be gathered until those decisions are being made. In this method, gathering is divided into "schematic facts, design development facts, and construction document facts" and it occurs WITH those respective synthesis stages.

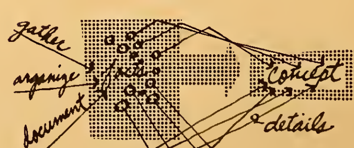


G. Where data gathering is integrated with design synthesis (immediate evaluation), the designer may take an ATOMISTIC (suboptimized) or WHOLISTIC approach.

1. In the ATOMISTIC approach, the programmer (who in this case is also the designer) tries to find optimal solutions to SUB-PROBLEMS or individual situations as they are uncovered in fact gathering. He later attempts to combine these "sub-solutions" into a coherent WHOLE without compromising them. This approach assumes that since a building "works" at this very specific level, the designer should begin with solving those problems first. It also holds the value that the whole is no more than the sum of the parts and that if all the specific aspects of the building are successful, the "whole" by definition will be successful.

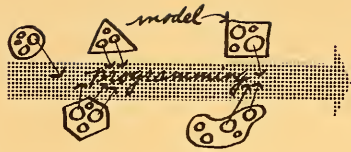


2. The WHOLISTIC approach subjugates "sub-solutions" to the larger context of a SCHEMATIC CONCEPT. Here a framework or overall organizational idea is established first and the more detailed concerns are "worked out" within the model. The "broad" concepts are determinants WITHIN WHICH the remainder of the problem must be



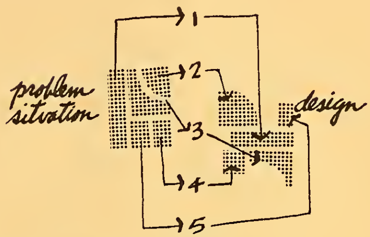
solved. For this reason, the sequence of information gathering is very important. That which is gathered and responded to FIRST sets the general direction for the solution.

H. The models discussed above may or may not occur in their PURE form. A programmer may use combinations and other models depending on his view of design. It is important to know the models to be used prior to planning the gathering procedures.

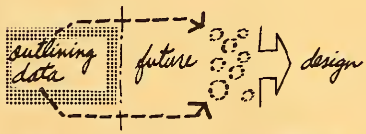


IV. OUTLINING DATA TO BE COLLECTED

A. It is in this stage of the programming that the ELEMENTS are identified that are to be MANIPULATED in design. The manner in which the facts to be gathered are IDENTIFIED and GROUPED begins to determine how the problem situation is "divided up" into manageable pieces which in turn influence the pieces which the DESIGNER will attempt to put together into some sort of coherent whole. It is important that the programmer be CONSISTANT throughout his entire process once the problem parts have been identified.



B. In the interest of efficiency it is of value to know what data is needed and what is not needed PRIOR to beginning data gathering. The cost of gathering unusable data is high and of evaluating, analyzing and organizing it even higher.



It must be recognized that with the EXPERIENCE that allows an efficient gathering operation also comes the danger of forcing NEW situations into OLD molds.

C. Fact gathering should NEVER be done "cold." Prior to outlining his facts to be gathered, the programmer should be as familiar as possible with:

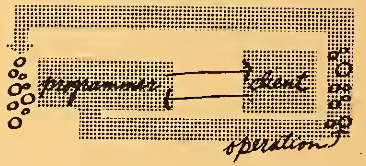
1. past solutions to similar design situations.
2. prevalent critical issues in the client's operation.
3. current trends and developments.
4. general problem areas encountered in the building type.
5. the terminology for communicating with the client about his operation.

In effect this amounts to "unofficial" fact gathering. Its purpose is to enable the programmer to be EFFECTIVE at his task. This familiarization or introductory involvement will help to avoid the "unusable data" problem and will facilitate the DEFINITION of that information which is crucial to the project.

D. Some of the WAYS that familiarization can be achieved are:

1. checking the art index for all articles on the building type including examples of past designs.
2. searching the libraries for books on the client's operation and the building type.
3. reviewing journals or other periodicals that specialize in the client's process.
4. contacting organizations that might supply literature on the client's operation.
5. writing for reports on conferences held on the subject.
6. writing prominent individuals in the field for a review of their current work.
7. compiling a bibliography from all the above sources and acquiring as many of the pertinent publications as possible.
8. visiting existing buildings which house similar functions and interviewing people there if possible.
9. attending conferences on the client's process or on planning for the client's process.
10. executing a quick design esquisse to identify what may be critical information areas or particularly difficult design problems.

Familiarization also permits the programmer to talk KNOWLEDGEABLY to his client about his operations. It should never be the client's responsibility to "educate" the programmer in the broad issues of his (the client's) field.



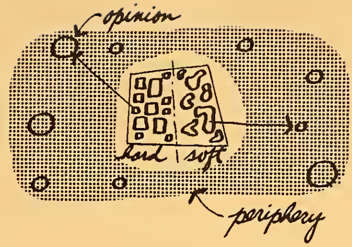
E. The TYPES of facts and the degree of DETAIL required may depend upon:

1. the purpose of the program document (for the client to make decisions?, to design from?, to feed into a computer?).
2. the degree of complexity, precision and size of the client's operation.
3. the performance standards required of the building.
4. the number of special or unusual conditions involved.
5. the nature of the project regarding new construction, addition, remodelling or a combination of these.
6. the values of the programmer as to non-traditional architectural facts and the level of detail he feels must be responded to in design if the building is to be successful.
7. the uniqueness of the project. The more "common" the building type the more the programmer may tend to "assume" that the designer knows about the client's process.
8. the philosophy of the designer.

F. Where the client is a LARGE organization intending to undertake a PHASED expansion project, there may be "pre-programming" data gathering to help determine the

NATURE and SCOPE of the first phases of design and construction.

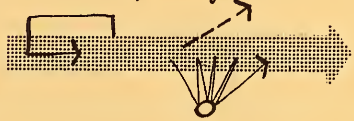
G. Some of the potential FACT CATEGORIES are outlined in the section on Traditional Architectural Facts. It is important to note that both QUANTITATIVE "hard" facts and QUALITATIVE or "soft" facts are needed. The program must give the designer a SENSE of the problem. This sometimes means that the program should contain a significant amount of the programmer's OPINION or information that he might ordinarily consider PERIPHERY.



V. DESIGN OF FORMS AND FORMATS

A. In gathering facts, especially for more complex projects, it is of value to RECORD the information AS IT IS GATHERED. Do not depend on remembering.

Without the documentation of the facts as they are gathered much of the programming effort can be WASTED in erroneous interpretations, retraced steps and multiple verifications.



B. The design of the FORMS on which data will be collected may be influenced by several factors:

1. THE TYPE OF INFORMATION TO BE GATHERED– Is it qualitative or quantitative? Does it lend itself to graphic or verbal representation? Does it involve a large number of people or other sources of just a few?

2. THE WAY THE INFORMATION WILL BE GATHERED- Will you gather it yourself or send assistants? Will an interviewer be present or will the subject simply fill out a questionnaire at their own convenience? Can the information be gathered at your own pace or must you record facts as fast as the client can talk?

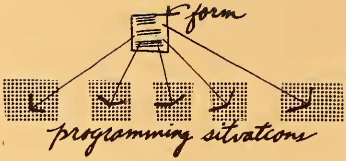
3. THE WAY THE GATHERED DATA IS TO BE USED– Can the gathering form also provide an opportunity to see relationships between facts? How can the gathering form facilitate evaluation, analysis and organization processes?

4. THE REUSABLE VALUE OF THE FORM– Is the subject matter standard enough that it could be used in a later job? Would the building of a "data bank" be of value (information from many separate projects for use in future similar projects).

5. THE RELATIONSHIP TO THE OTHER FORMS– Will

all the forms be grouped to form a raw data "package"?

C. The forms used to GATHER data are very strongly related to those used for EVALUATING, ANALYZING, and ORGANIZING information after it is collected. Firms that are active in programming ordinarily develop STANDARD forms for gathering their information. Some of these include:



1. functional matrices.
2. sensory production - conflict matrices.
3. function - context matrices.
4. critical path diagrams.
5. site evaluation forms.
6. questionnaires.
7. drawings of plans for existing buildings housing client's operation.
8. checklists.
9. bubble diagrams of affinities, conflicts and sequences.
10. furniture inventory forms.
11. specific space needs form (furniture, electricity, HVAC)
12. code check form.

Other FORMS used for collecting data are tape recorders, photography, sketching, xerox and game playing.

VI. DEFINITION OF SOURCES AND EXECUTION

A. For each "bit" of information outlined as being needed by the programmer, he must also know WHERE he can get that fact. This awareness is actually needed BEFORE he can plan his procedure for collecting his data.

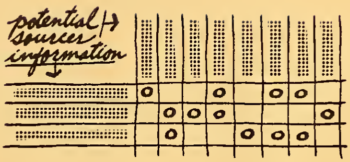
B. Typical "source areas" with which the programmer may be involved in gathering facts are:

1. interviews with the client himself.
2. interviews, questionnaire surveys and observation of the client's staff and operation.
3. consultants (site surveys, soil tests, furniture and equipment, efficiency experts, researchers, electrical, mechanical, structural, fund raising, financial planners).
4. books and periodicals on planning for the building type.
5. general planning standards (FHA Minimum Property Standards, Time Saver Standards, Building Planning and Design Standards, Graphic Standards).
6. planning information from pertinent associations and manufacturers.
7. Uniform Building Code and local zoning ordinances.
8. governmental regulations.
9. empirical measurement of important sensory situations.
10. manufacturers' catalogs and representatives.

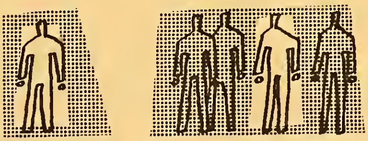
11. city building inspector.
12. city planning and utility departments.
13. local utility companies.
14. local aerial photography firms.
15. city, county and state studies and publications (population growth, traffic volume, visual surveys).
16. studies done by local firms such as banks or utility companies (projected growth, etc.)
17. books and publications on cost data ("Construction Outlook" — F. W. Dodge, "Dodge Building Data and Cost," "National Construction Estimater").
18. subscription to services such as "IDAC," "Pattern Language," or "CAD-LAB."
19. weather bureau.
20. personal visits and observation.
21. school district surveys and publications.

Some of the "methods of familiarization" listed earlier also apply to this concern.

C. It is often helpful to list ALL the potential sources for EACH fact needed. This fact-category-potential source matrix is very useful where there is more than one gatherer involved. Tasks can be easily divided among the workers. The matrix can be DEVELOPED and EXPANDED as it is used again and again for different projects.



D. Don't overlook YOURSELF as a principle source of information. It is usually quite effective to "empty your head" in writing regarding EVERY issue that may have an IMPACT on the problem. These may in turn be organized into topics and sub-topics. BRAIN STORMING with fellow programmers may add to the issue list.



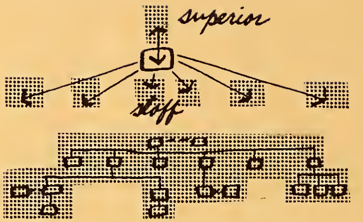
E. In actually executing the gathering of the information there are several factors that may be influential in having the gathering operation succeed.

1. When interviewing the client or his staff:

   a. try to avoid SPONTANEOUS meetings or phone calls.
   b. attempt to plan meetings and set up appointments for SPECIFIC purposes.
   c. have an agenda and avoid tangents except where necessary.
   d. try not to OVERSTRUCTURE an interview. Allow the client freedom to communicate. Often, the client's initial comments regarding what he feels are important issues prove to be major determinants. The client should be allowed to express these at the start of an interview.
   e. client comfort, attention span, boredom, participation

and involvement are key issues when interviewing.

f. attempt to get the client to quantify his qualitative statements wherever possible ("on a scale of one to ten"). This will provide a clearer understanding of relative values he places on his needs.
g. have the client talk in terms of NEEDS and not solutions.
h. where administrative commitments need to be made before you can continue with programming, outline the situation but let the client make the decision. Always have client representatives present who have the authority to make decisions that won't be changed by superiors.
i. always VERIFY data collected in interviews by writing reports of the meetings and sending copies to all concerned.
j. when interviewing staff, always touch base with their administrative superiors. Staff can define needs but administration must have the final decision as to the degree to which those needs will be satisfied.
k. know the decision-making structure of the organization. Where appropriate, have the client designate a committee to work with you. Be sure of their decision-making responsibilities.



2. When using a survey or questionnaire to be executed by staff without supervision:

    a. attempt to explain the form personally to all involved.
    b. include an explanation sheet telling the PURPOSE of the survey as well as giving instructions for executing it.
    c. try to avoid any ambiguous questions. Whenever possible judgments of those surveyed should be expressed QUANTITATIVELY.
    d. relate the survey results to those who were involved. Their understanding of the value of their efforts is important to securing their continued cooperation.

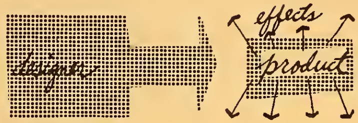
3. When using a consultant, always be very clear and EXPLICIT about WHAT YOU WANT THEM TO DO and the form in which you expect their findings.

4. As in design, the programmer's sensitivity, awareness and analytic-synthetic abilities are CRITICAL to his success. Programming is not a mechanical endeavor but still largely an ART where creativity, initiative and a constant search for new ideas are VITAL.

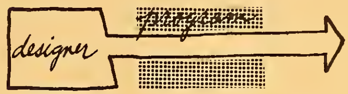
ANALYSIS, EVALUATION AND ORGANIZATION OF FACTS

I. CONTEXT

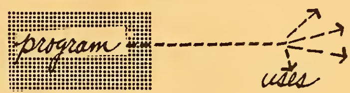
A. Design synthesis involves COMMITMENTS made by the designer. He must ADVOCATE, PROPOSE and RECOMMEND and finally make relationships between particular and individual elements so that the effects of his product are as anticipated.



The architectural program STRUCTURES, LIMITS, DIRECTS and DEFINES those commitments and the making of relationships. The program is the "plan of proceeding with synthesis."



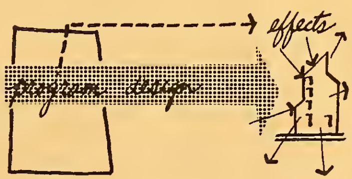
B. The architectural program is never an END in itself but an instrument to be used in some SUBSEQUENT process. Its uses and roles must be KNOWN to insure that it be made a usable and effective working tool.



C. Analysis, evaluation and organization of facts are ESSENTIAL to the making of an effective and usable program.

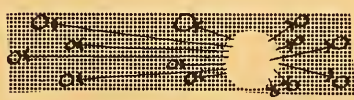
II. GENERAL CONSIDERATIONS

A. Both design and programming involve PREDICTIVE techniques. The program is concerned with defining building CONSEQUENCES which are considered desirable and directing the design process to bring these INTENTIONS to REALIZATION. Analysis, and organization of information in programming are meant to SUPPORT these goals.



B. Definitions

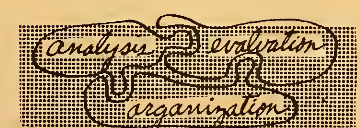
1. Analyze: To separate or break up a WHOLE into its PARTS so as to find out their nature, function and relationships; examination of the relations between variables.



2. Evaluate: To determine RELATIVE importance; to appraise.

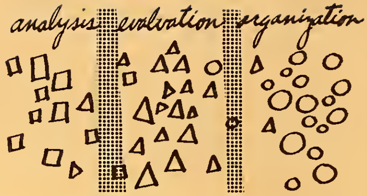
3. Organize: To STRUCTURE, arrange, establish or order.

C. The actions defined by these three terms are very individual and specific. It is impossible, however, to COMPARTMENTALIZE each operation separate from the other two in programming. Evaluation is needed in both analysis and organization. There must be some organization for

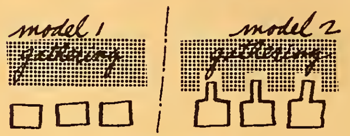


evaluation and analysis. Analysis provides evaluation with subject matter.

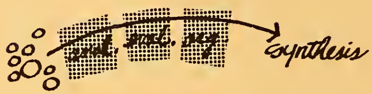
Like the phases of the total design process (programming, schematics, design development), "analysis", "evaluation" and "organization" only identify CONCENTRATIONS of similar kinds of activity that in effect permeate each other to differing degrees. They are separated as operations here not to propose that they actually occur as distinct and separate packages but to study and hopefully refine and improve them.



D. Whether these processes are considered INTEGRAL or SEPARATE from data gathering depends on the "models" that we use for gathering (separate versus integral gathering with synthesis, immediate versus deferred evaluation of facts).



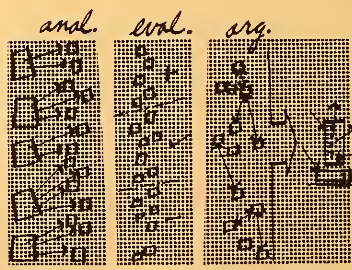
E. ANALYSIS, EVALUATION and ORGANIZATION must bridge the gap between RAW data and DESIGN SYNTHESIS. Out of these processes comes the material for the production of the program document.



F. The FORM in which the data comes from GATHERING may facilitate or hinder these processes. It is of value to perform as FEW operations on the form of gathered data to make it USABLE for analytical, evaluative and organizational tasks as possible.

III. ANALYSIS OF FACTS

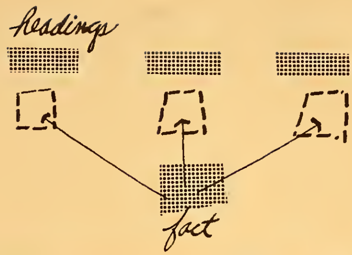
A. In ANALYSIS, the programmer is principally interested in the DECOMPOSITION of the data into its components. The process plays a supporting role to evaluation in that facts are broken down to allow very SPECIFIC and DETAILED determination of relative importance. Analysis also is important to organization because it uncovers RELATIONSHIPS between facts and between facts and building consequences. Qualities of facts that establish similarities and differences are determined in analysis. These qualities are used as a BASIS for SORTING and GROUPING of facts into SYSTEMS.



B. The decomposition of information or facts into smaller comprising SUB-ISSUES not only serves to reduce the data to a "finer grain" which can be more easily dealt with in design but also often results in the UNCOVERING of what prove to be major design determinents which otherwise might have remained BURIED within broader more general facts.

C. If each design issue or fact category is EXHAUSTIVELY

extended with respect to ALL related subissues and sub-subissues, there will be considerable OVERLAP and REPETITION regarding the fine grain information. The same bits of information will be claimed by different issue headings. The resolution of this problem must occur in the ORGANIZATIONAL processes of programming. After analysis, fine grain information may be GROUPED and ORGANIZED totally differently than when the process began and NEW topic headings may need to be invented for the new information groupings.



D. Analysis does not finally fix the relationships between data that will be used in synthesis. It is NOT a synthesis operation but deals only with discovering POTENTIAL relationships and qualities for information, organization and design.

E. Some of the qualities and relationships that offer potential means for organizing the data are:

1. the types of CONSEQUENCES that the fact deals with (social, economic, physical, psychological)

2. the ELEMENTS of the future building, the design of which must respond to the fact (site, structure, function, environment)

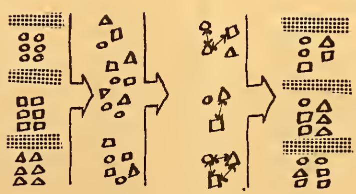
3. the relative IMPORTANCE of the fact to the client or to the designer

4. the SEQUENCE in which the facts will be used in synthesis (schematic, design development)

5. the relative FLEXIBILITY or FIXEDNESS of the fact (hard versus soft data)

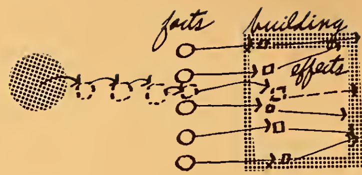
Also of use in the analysis of facts are those qualities that result from the EVALUATION of the data.

F. The importance of analysis as a separate operation will depend on how STRUCTURED the gathering process has been in terms of FACT CATEGORIES. Even where the relationships between data have been predetermined for purposes of convenience and efficiency in gathering, it may sometimes be valuable to DECOMPOSE the "fact organization" to provide an opportunity for discovering NEW and CREATIVE potential relationships between facts.

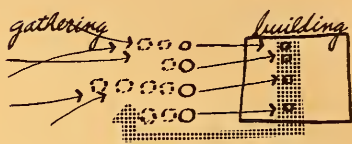


G. Analysis is directly concerned with the study of specific facts in terms of their POTENTIAL IMPLICATIONS on the physical building. As in "non-traditional facts", these

implications are sometimes not immediately evident. The programmer must be perceptive and thorough enough to trace the implications of even seemingly "remote" data on the building design. Remoteness of a fact as a causative agent to the surface event does not mean it is not relevant, that is, part of the chain of events leading to the BUILDING CONSEQUENCE.



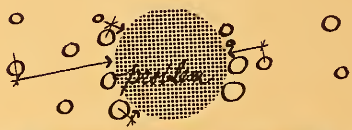
H. An important by-product of checking facts for their architectural implications is that it may point out the need for MORE DATA in certain areas. This feedback to gathering also results in refinement of gathering techniques.



IV. EVALUATION OF FACTS

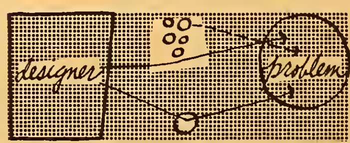
A. Evaluation here is to be DISTINGUISHED from the evaluation of fact relevancy in data gathering or the appraising of design decisions or final building.

B. As facts are gathered (or after they are "all" gathered and analyzed), their RELATIVE IMPORTANCE to the problem must be determined. The programmer must have some bases or criteria for making these judgments. The criteria for deciding the relative importance of data may relate to:



1. Whether the data has a DIRECT bearing on the design of the building or not.

2. Whether the fact, need or future desirable situation is one that will be AUTOMATICALLY taken care of by the solving of other problems, response to other problems, response to other facts or satisfaction of other needs or whether it demands the DIRECT attention of the designer.



3. How SOON the fact will be important to the designer's operation.

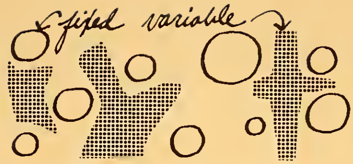
4. The relative IMPORTANCE of the fact in terms of the client's goals.

5. The relative importance of the fact in terms of the goals of the ARCHITECTURAL firm.

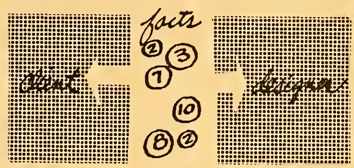
6. The relative FLEXIBILITY or FIXEDNESS of the fact, ("hard" or "soft" data) .

C. Through evaluation, PRIME ORGANIZERS are identified which may serve in the forming of CONCEPTS in synthesis.

Also, by defining the facts that are fixed and unchanging (site shape), the designer is made aware of the FRAMEWORK around which the more VARIABLE aspects of the problem must be worked. The GREATER the body of fixed facts, the FEWER the alternative solutions that will be available in synthesis.

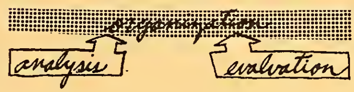


D. Where a large number of facts are involved, it is sometimes helpful to assign QUANTITIES to the criteria for evaluation and to express the relative importance of the facts NUMERICALLY. This promotes clarity in the feedback to the client and in the communication to the designer regarding the VALUE RANGE assigned to problem determinants.



V. ORGANIZATION OF FACTS

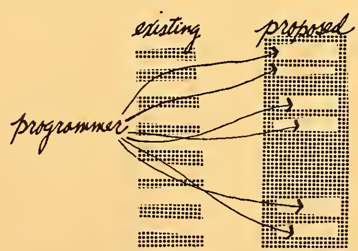
A. Analysis and evaluation are TOOLS of the organizational process in programming. Both are necessary as BASES for organizing program information.



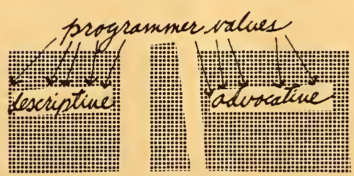
B. Although there is a degree of organization related to both analysis and evaluation, organization as a FORMAL process in programming usually happens AFTER the data has been evaluated and analyzed. This is true even where analysis and evaluation are integrated with the gathering process.

The work done in analysis and evaluation should be REFLECTED in the organized data.

C. Organization is the SYNTHETIC, DECISION-MAKING operation in programming. Here the programmer begins to make COMMITMENTS in terms of relationships and qualities to be used in design. He begins to draw conclusions and make recommendations about what should happen in schematic design and design development. Involvement extends BEYOND a description of the "existing" to a projection of future desirable situations. The program should contain statements about HOW this might be achieved.

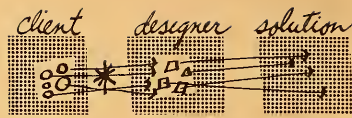


It is advisable for the sake of clarity that DESCRIPTIVE statements about the existing situation and ADVOCATIVE statements about what SHOULD happen be DISTINGUISHED in the program. Even though all of programming reflects the values of the programmer, statements that are obviously judgemental should be clearly indicated as such.

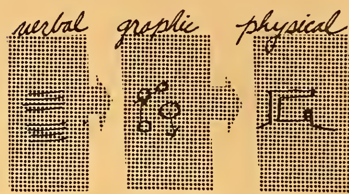


D. The organization of data is the essential process for bridging the gap between the PROBLEM STATEMENT

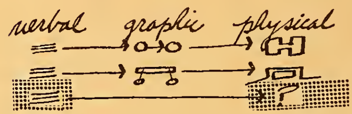
and the SYNTHETIC OPERATION that will result in a solution. It is the point where client needs and their relationships with the other facts gathered, analyzed and evaluated are TRANSLATED into the language of the designer.



Needs and other facts at the gathering stage are largely VERBAL concepts. As architecture is a PHYSICAL (visual) expression of the solution to the problem statement it is of value to express as much of the program GRAPHICALLY and DIAGRAMMATICALLY as possible. This diagrammatic translation of the programming facts is the start of the formation of the physical building, as diagrams have DIRECT implications on physical building form.



The programmer's ability to design visual, graphic communication of programming data will largely determine the extent to which all the programming NEEDS are met in synthesis.

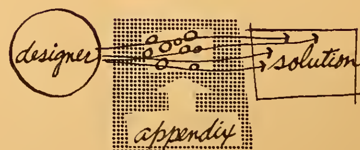


E. The SEQUENCE of data and the FORM in which it appears must be related to the WAY it will be used. Ideally, after the program is complete, there would be no additional operations performed on the data to make it directly usable in design. This may sometimes create difficulty as the same facts often should take DIFFERENT forms when being used for DIFFERENT design tasks.

F. It is helpful to the designer if the program format CLEARLY indicates information types, priority and emphasis. Some of the WAYS that may be used to communicate these issues are:

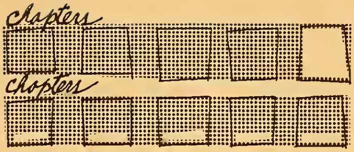
1. diagrammatic expression of important issues
2. use of capital letters, italics or underlining words
3. tones applied over important phrases
4. color coding of title pages or pages of a section
5. use of large dots or other shapes beside important facts
6. use of receding page sizes to reveal all program sections simultaneously
7. tabs applied to each program chapter or section
8. tables of contents at each chapter in the program

G. As a DESIGN INSTRUMENT, the program should be organized to allow the designer to easily FIND and USE data that is directly pertinent to synthesis. A common response to this need is to group all supporting information in an APPENDIX, separating it from the facts that have DIRECT architectural implications.



H. The use of SUMMARY SHEETS where all critical data

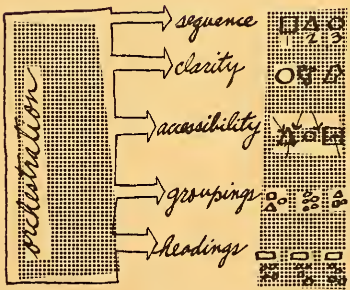
under a given heading is GROUPED and SUCCINCTLY presented is of great help to the designer. Typical data sheets might include space needs, code requirements or overall functional relationships. These may be grouped together in a summary CHAPTER or may occur separately in EACH topic section.



I. Related to the summary sheet concept is the issue of STANDARD information forms. Highly systematic programming may involve gathering the information on these forms with the saving of hours of organization time later. SPACE ANALYSIS summary sheets are the most common of these forms.

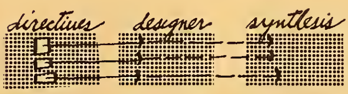
J. The major information HEADINGS that proved useful in gathering, analyzing and evaluating the information may or may not CONTINUE as the major headings in the organizational processes. After decomposition of data in analysis, it may be REGROUPED on the basis of newly discovered SIMILARITIES and DIFFERENCES. Totally new information groups and titles may emerge which have little relationship to those used for the earlier tasks.

K. From the preceeding issues it becomes clear that the ORCHESTRATION of the data (sequence, clarity, accessibility, groupings, major headings) is as important as the INFORMATION itself. A very strong determinent is the particular manner in which the elements to be ASSEMBLED in design have been IDENTIFIED. In putting a building together, the designer may work in any of several ELEMENT SYSTEMS (people, activities, room areas and shapes, space volumes, furniture). The information groupings and their titles establish a VIEW of the problem that strongly promotes the use of CERTAIN element systems over OTHERS.

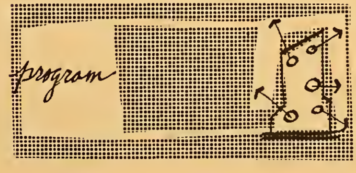


L. As in the gathering of information for programming, organization may be based on a MODEL or concept about its RELATIONSHIP to the design of the final building product. Two such examples are:

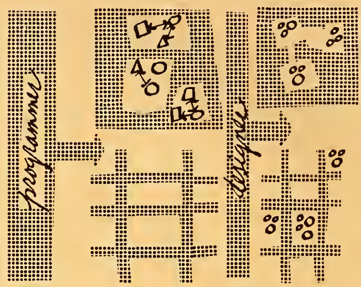
1. THE PROGRAM IS A SET OF INSTRUCTIONS TO THE DESIGNER. This implies that the program format take the form of a series of DIRECTIVES.



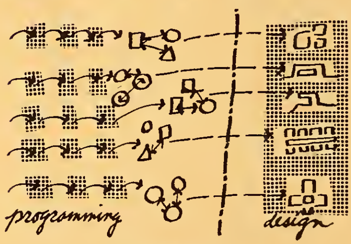
2. THE PROGRAM SHOULD DESCRIBE THE FINAL DESIGN AS EXPLICITLY AS POSSIBLE IN VERBAL AND DIAGRAMATIC TERMS. This involves not only drawing conclusions about the consequences that individual aspects of the building should have but also PROPOSING the physical building situations that will most effectively bring them about.



M. The programmer should not be concerned about INFRINGING upon the PROVINCE of the designer. The LINE between programming and design operations is in DIFFERENT places depending on the project issue involved. Different people may have differing opinions on the matter also.



The program should contain INTERPRETIVE information that refers to the ARCHITECTURAL IMPLICATIONS of the raw data. The programmer's preferences for directions in design should be clearly indicated. This tactic provides the designer with the recommendations of those MOST FAMILIAR with the problem. It is always the designer's OPTION to ignore the design content in the program. The extent of the design content in the program is up to the programmer. Some may stop at suggested SUB SOLUTIONS with the designer assembling these into a WHOLE. Others offer concept FRAMEWORKS within which the designer works out the DETAILS.



The fundamental PREMISE behind this attitude is that it seems UNREASONABLE to develop information from its raw state through several stages of translation to its architectural implications and then to terminate the process at some IMAGINARY and ARBITRARY line between programming and design operations. It seems much more reasonable to CONTINUE the process to its CONCLUSIONS, allowing the designer to CHOOSE how much of the design content he will use.

N. Depending on the nature of the project, it is often desirable to have DESIGN DEVELOPMENT information available when doing SCHEMATIC DESIGN. This permits the designer to TEST his schematic design decisions against the more detailed requirements that schematics must eventually ACCOMMODATE. Schematic design can proceed more CONFIDENTLY with a view toward what is TO COME.

O. In outlining a program for schematic design, the inclusion of what might ORDINARILY be considered "details" can serve two purposes.

1. For the value of the INFORMATION itself as a PREVIEW of requirements which must eventually be met.

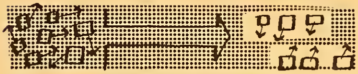
2. As a CATALYST for discovering what may prove to be SCHEMATIC DESIGN issues.

P. Like the analytical and evaluative processes, the operations performed on data in organization depend on HOW MUCH was done to the data during its gathering. Some

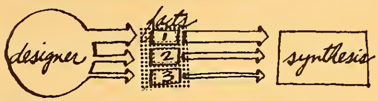
example organizational operations are:

1. SORTING and GROUPING of facts into categories based on qualities identified in analysis and according to criteria established by the programmer (sequence of use, relative importance).

2. Sorting and grouping of the EFFECTS on the design of individual building aspects implied by the program data.



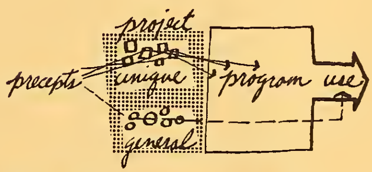
3. Establishing a HIERARCHY of determinants which will direct the sequence and intensity of the designer's attention in synthesis.



4. Writing DEFINITIVE precepts describing individual conclusions about the data and proposals about what the final design should accomplish.

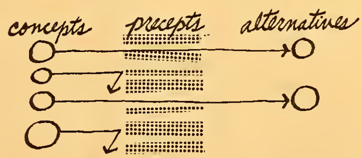
   a. Precepts should be SHORT, CONCISE, deal with only ONE issue at a time and be expressed GRAPHICALLY.

   b. Precepts should identify the UNIQUENESS of the problem. The extent to which general or "universal" precepts are written down and contained in the document depends on the PURPOSE of the document. OBVIOUS precepts may need to be included when EDUCATING the client.

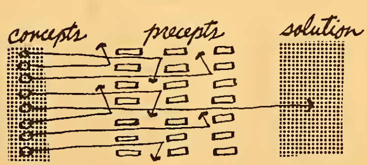


   c. Precepts should deal with issues involving building SECTION and ELEVATION as well as plan. This will help to avoid the "extruded plan" difficulty.

   d. An important role of precepts is that of EVALUATORS of directions taken in the conceptualization stages of synthesis. By checking alternative design directions against the precepts, the development of INVIABLE concepts can be avoided. Precepts help SCREEN and EVALUATE design alternatives.



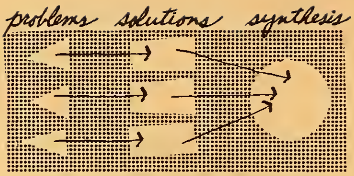
   Theoretically a comprehensive establishment of precepts at all levels of design synthesis (schematics, development) will result in a CONVERGENCE to the most viable solution to the problem. Hence, the statement, "the solution is contained in the statement of the problem."



   e. The use of precepts can help identify POTENTIAL CONFLICTS in the design problem. This is most clearly illustrated when two precepts COMPETE for a response from a particular building aspect or

element where a response to one EXCLUDES the possibility of responding to the other.

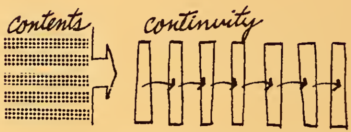
f. PATTERN LANGUAGE (Alexander) is closely related to the precept model. Essentially it proposes synthetic solutions to sub-problems which can be used in designing many different building types. The RESOLUTION of conflicts in the patterns and the SYNTHESIS of them into a whole is left to the DESIGNER.



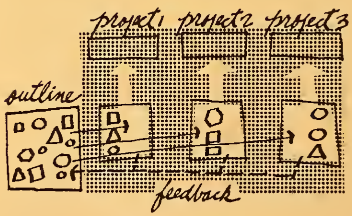
5. Identifying the ALTERNATIVE CONCEPTS for the design of the building SUGGESTED by the precepts.

6. Putting all the analyzed, evaluated and organized data into USABLE form (presentation). This task has special implications where the program is to be published or where data is to be fed to a computer for sorting or grouping.

Q. Oftentimes the discipline of designing a document for LOGICAL CONTINUITY can be of help in structuring the organizational processes in programming. In a sense this is designing the program through designing its table of contents.



R. One programming tactic that often proves useful is the development of a reusable PROGRAM OUTLINE. As it is used from project to project it may be EXPANDED and REFINED. A comprehensive program outline is usually never COMPLETELY applicable to every project. It must be TAILORED to suit the building type under study. An outline can serve as a CHECKLIST to insure a thorough and organized programming effort.



S. A program outline should not only be as DETAILED as possible but it should also convey a sense of information PRIORITY with respect to schematic design and design development.

T. There are several considerations that may assist in the development of a program outline.

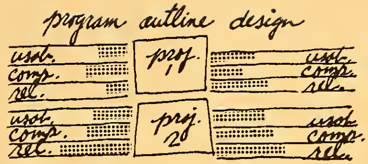
1. COMMITMENT TO YOUR VIEW OF DESIGN. A design view or way of understanding and explaining the design process can help in firming up views about programming and its role in that process.

2. READINGS IN PROGRAMMING. Familiarity with historic and current issues in books, periodicals, conference papers and publications of professional firms provides a base for forming a personal programming approach.

3. REVIEW AND ANALYSIS OF SAMPLE PROGRAMS. It helps to see how others have structured their programming approach and the information types that have been used by different firms for different building types.

4. PRELIMINARY PROGRAM OUTLINE. The first attempt at the outline should be as organized and detailed and usable as possible. "Emptying your head on paper" is a good way to start.

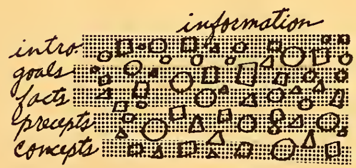
5. BUILDING PROGRAM AND DESIGN. The outline must be tested for usability, comprehensiveness and relevancy to both the programming and the design tasks. On the basis of as many of these applications as possible, the outline can undergo many evaluations and refinements.



6. DESIGN EVALUATION. It is often revealing to use the building program as criteria for evaluating the design. The degree of applicability of the program in this role many times provides insight into needed outline alterations.

A program outline probably never reaches "final form." It must be continually USED, EVALUATED and IMPROVED.

U. Ordinarily the MAJOR program subsections of INTRODUCTION, GOALS, FACTS, PRECEPTS and CONCEPTS adequately serve as DIVISIONS of programmatic information. In organizing a SPECIFIC program however there is often a need to TAILOR the information groupings to suit the UNIQUENESS of the project.



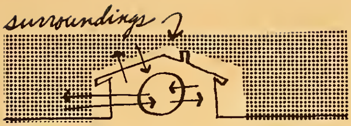
Some of the information CATEGORIES or program SUBSECTIONS are listed below. They are not ordered in any particular manner but are intended only to briefly present the scope of AVAILABLE titles which may be used in organizing a program. The CHOICE of these titles and their ARRANGEMENT in a program would depend upon the overall ORGANIZATIONAL CONCEPT of the document. It should be noted that there is some overlap between this list and the outline of traditional architectural facts.

1. pre-programming
2. acknowledgements
3. forward or preface
4. table of contents
5. purpose of the document
6. scope of the document
7. spirit of the problem (quotes)

8. client identification
9. client background and philosophy
10. history of client operations
11. general client goals
12. goals of specific project aspects
13. general trends in client's field
14. glossary of client vocabulary
15. time schedule and budget
16. project priorities
17. program organization and format
18. programming methodology
19. overall project goals and objectives
20. project status
21. project descriptions
22. reason for the project
23. general design philosophy
24. general description of client's operation
25. major constraints and limitations
26. analysis of existing conditions
27. facts (see Traditional Architectural Facts)
28. precepts - general explanation
29. site precepts
30. building precepts
31. phasing precepts
32. premises
33. assumptions
34. givens
35. architectural design criteria
36. general building systems design criteria
37. mechanical systems design criteria
38. electrical systems design criteria
39. structural systems design criteria
40. building performance (consequence) standards
41. concept alternatives
42. patterns
43. action plan
44. concept aspects (description)
45. evaluation of concepts (advantages and disadvantages)
46. composite evaluation
47. project phasing
48. recommendations
49. review
50. general conclusions
51. summary
52. appendix
53. exhibits
54. definitions and glossary
55. index
56. bibliography
57. credits and programming team

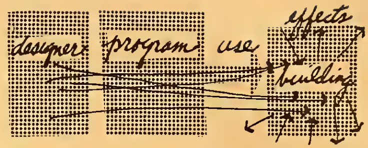
V. All of the above information types INFLUENCE the nature

of the CONSEQUENCES that the resulting BUILDING will have on its SURROUNDINGS and CONTENTS and that its SURROUNDINGS and CONTENTS will have on the BUILDING.

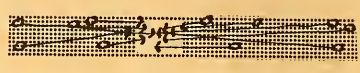


DESIGNING FROM THE PROGRAM

I. GENERAL CONSIDERATIONS



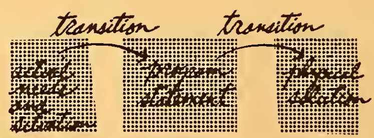
A. Although its ROLES may vary, the principle purpose of a building program is that of a DESIGN TOOL. Its validity lies in its USE and its value depends on the degree to which it facilitates the synthesis of a building design solution that is successful in all its predicted and desired CONSEQUENCE aspects.



B. Synthesis: the putting together of parts or elements so as to make a WHOLE.

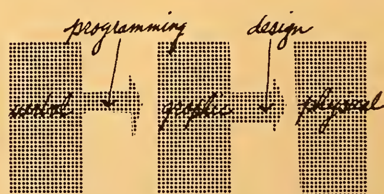
C. As a "design event" the nature of the response to the program in synthesis depends largely upon HOW the programmer gathered, analyzed, evaluated and organized the information.

D. Depending on the amount of SYNTHESIS already contained in the program, the "parts" to be assembled in design may range from a simple statement of desired consequences with no stated architectural implications to a series of presynthesized sub-solutions such as pattern language or precepts that describe optimum ARCHITECTURAL responses to individual needs.



E. In the same way that programming is a transition from the ACTUAL client needs and the ACTUAL existing situation (site, climate) to an organized statement to DESIGN BY so also is synthesis a transition from the program STATEMENT to the actual PHYSICAL solution.

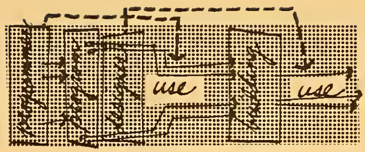
Both programming and synthesis can be thought of as TRANSLATIONS where a situation in one language is expressed in another.



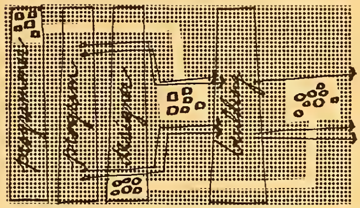
The programmer takes the "raw situation" and TRANSLATES it into the language of the designer. The designer in turn TRANSLATES the program into a physical solution. The first expresses VERBAL concepts GRAPHICALLY, the second expresses GRAPHIC concepts ARCHITECTURALLY. If a concept cannot be expressed GRAPHICALLY, it usually cannot be expressed ARCHITECTURALLY.

For the building to accurately and comprehensively express the original "raw situation" that initiated the entire process, BOTH translation operations are critically important. The INTENT and MEANING of the original

situation must be presented in both programming and synthesis.



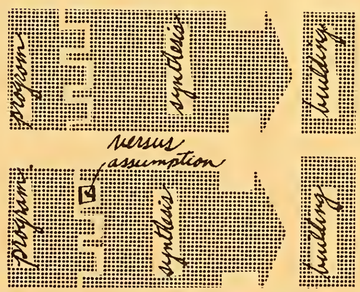
F. As the DESIGNER must anticipate and simulate the use of his building to insure that it functions to suit the future situations, so also must the PROGRAMMER anticipate and simulate the use of his program to insure that it functions successfully as a tool in synthesis.



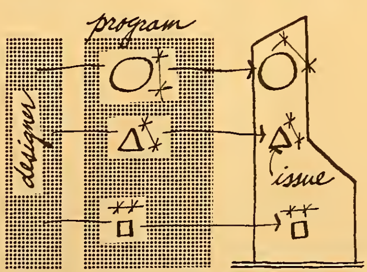
The simulation required for both depends upon previous experience (direct or learned) in situations similar to those yet to be. This need is more commonly recognized when designing the building than when programming, yet no more important. SYNTHESIS may fail due to poor PROGRAMMING, just as the BUILDING may fail due to poor SYNTHESIS. The net result of either is a building that does not successfully respond to the original situation which was brought to the architect by the client.

II. PROGRAM - DESIGN RELATIONSHIPS

A. There are several QUALITIES of the program-synthesis relationship that are of value:



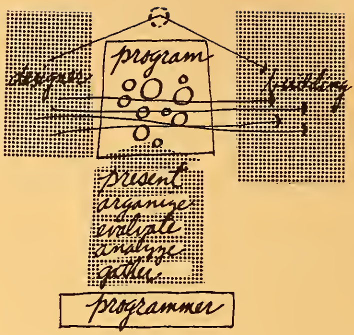
1. THERE SHOULD BE MAXIMUM INTERFACE BETWEEN PROGRAM AND SYNTHESIS. Ideally, the planning process should be CONTINUOUS from the original situation to the realization of the building. The program should DETERMINE the solution. Synthesis should be directed as completely as possible by the program, and there should be no gaps between programming and synthesis to be "filled in" by the designer's "assumptions." If the program has clearly identified the ELEMENTS to be MANIPULATED in DESIGN and the issues involved in determining their relationships, design-program interface will be facilitated.



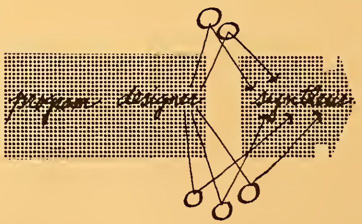
2. SYNTHESIS SHOULD BE FAITHFUL TO THE PROGRAM. Sometimes when manipulating the elements of the physical building, the designer may be tempted to INVENT new needs, INFLATE the importance of a determinant or DE-EMPHASIZE a critical issue to facilitate the resolution of some geometric, spatial, structural or aesthetic problem. Recognizing that ARCHITECTURAL (physical) concerns may sometimes cause a deviation from program intent, the designer should nevertheless strive to make his building an accurate reflection of the program statement.

3. SYNTHESIS SHOULD THOROUGHLY RESPOND TO THE PROGRAM. Some programs leave more for the

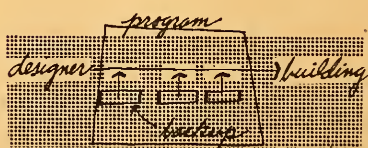
designer to "fill in" than others. The degree of detail and thoroughness required in synthesis is not optional to the designer but determined by the LEVEL OF DETAIL at which the building will function when occupied and in use. The designer may sometimes be inclined to cut short his development of the solution when it reaches the tedious stages of providing for the fine details of function. This thoroughness and attention to detail can be facilitated by the program.



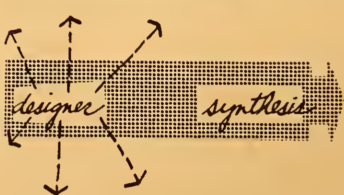
When the program is INCOMPLETE either in terms of general issues or details, unwarranted pressure is put on the designer to gather, analyze, evaluate and organize the needed information. When the designer must by-pass programming and go directly to the "original situation," there is a danger that the solution will begin to be "patched together." Ordinarily, the designer is concerned with "putting the building together" and will seldom do justice to the raw information in terms of its analysis, evaluation and organization prior to responding to it in his design. Here, all the potential IMPLICATIONS and RELATIONSHIPS that might have been discovered through reflection upon analysis of program information are lost.



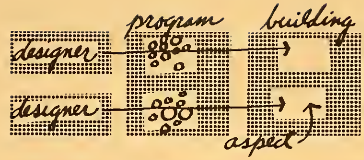
4. SYNTHESIS SHOULD FLOW UNINTERRUPTED FROM THE PROGRAM. When the designer must stop synthesis to gather more data or to translate it into usable form, this results in an INEFFICIENT and UNSYSTEMATIC response to the program. Where programming is PHASED so as to provide only enough data for a given segment of synthesis (schematics), that phase of synthesis should be able to be completed SMOOTHLY with the data supplied in the program.



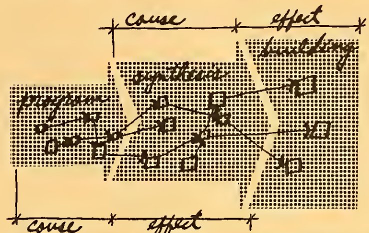
The separation of information that has DIRECT architectural implications from SUPPORTING or backup information allows the document to be much more efficiently used by the designer. An APPENDIX should be used for supporting information while directly usable data should be grouped and identified.



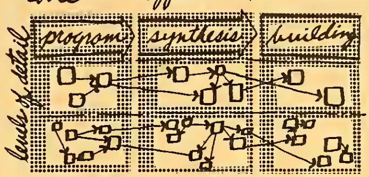
Anything that causes the designer's attention and concentration to be DIVERTED from synthetic issues is detrimental to the design process. The designer's "incubation," subconscious problem solving and creativity, even when not at the drawing board, should not be cluttered with thoughts relating to what must be done BEFORE he can begin designing (gathering more data, sorting out usable data).



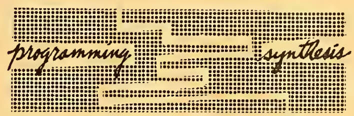
B. Where there is MORE than one designer on a project and different aspects of the design will be addressed by DIFFERENT people, in order to achieve the above mentioned quality the program must respond to multi-designer situations.



C. The view of "programming as a determinant of synthesis" and of "synthesis as a determinant of the building" are GENERAL descriptions of two cause-effect systems. However, the DETAILS of each system must be studied for the two systems to be OPERATIONALLY meaningful. SPECIFIC aspects of programming affect SPECIFIC aspects of synthesis and SPECIFIC aspects of synthesis affect SPECIFIC aspects of the building design.



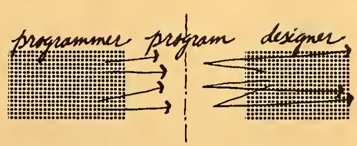
The isolation of specific cause-effect relationships between program and synthesis and between synthesis and building permits us to REFINE and IMPROVE both systems in a way that affects what the programmer and designer DO. This refinement cannot occur as long as relationships are studied on a general level remote from the actual particular operations of programming and design.



D. It is virtually impossible to precisely define a point where programming ENDS and synthesis BEGINS. The definition of programming as SEPARATE from design serves only to organize the fee structure in the profession and to identify and group operations of similar nature.

The "formal" beginning of building design is in the ORGANIZATIONAL process of programming. Depending on how far this process is taken the program will contain varied degrees of synthetic decision-making.

E. The stronger the DISTINCTION between programming and design, the greater the chances that the spirit of the program will be lost in synthesis. The one process should be CONTINUOUS with the other. This implies that the optimum situation in this regard is for the programmer to also be the designer. This, of course, assumes that it is of value for the designer to respond to all the subtleties of the program and the way the problem was understood in programming.

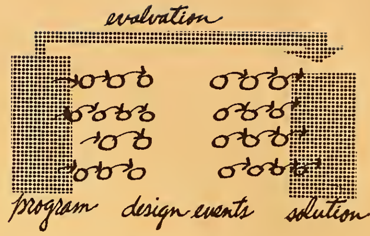


F. The most CRITICAL TEST of the communicative value of a program is where the programmer is not the designer and where the designer's ONLY exposure to the project is through the program document.

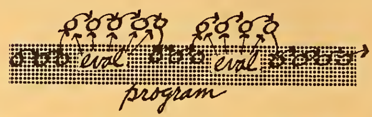
G. Where synthesis is CONTINUOUS with programming the term "response" is a misnomer. "Response" implies that

there is an INTERRUPTION in the continuum from program to design and .that they are two independent operations that are "brought together" artificially.

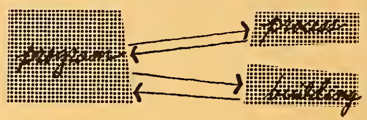
In the same way, the use of the program as a means of evaluating the final solution means little when the program and design are CONTINUOUS (high percentage of interface). If the solution is DIRECTLY generated by the criteria for evaluation, the design is by definition successful. Where the "stream of design events" between program and solution has been BROKEN, the use of the program as a criterion for evaluating the building becomes a more appropriate process.



Where the designer works on design independently of the program for periods of time, the program may also serve as an evaluator of INDIVIDUAL design decisions (decisions and directions tested against precepts).



H. As the program may be used to evaluate both the final DESIGN and the PROCESS leading to it, both of these may be used to evaluate the PROGRAM. Some of the ways that synthesis may test programming are:



1. degree of "fit" between the program and the designer's view of design (can he relate to it PHILOSOPHICALLY and OPERATIONALLY?)

2. thoroughness and required degree of informational detail

3. usability of the information forms and convenience of the overall format

4. relevancy of the data

5. information sequence as presented in the program

6. "visual palatability" of the document including the efficiency with which the designer can grasp program issues (related to strength and clarity of graphic expression of issues)

7. degree to which program serves as a catalyst in determining initial design concepts and directions

8. clarity of the priorities in the program as criteria for resolving design conflicts

9. degree to which program removes the need for arbitrary assumptions and judgments in synthesis

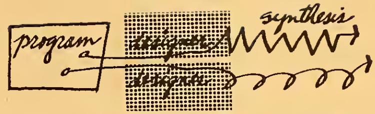
10. extent to which the program promotes a creative syn-

thesis of the problem elements and issues

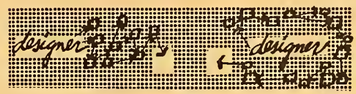
These criteria of evaluation may apply to ANY or ALL of the gathering, analysis, evaluation and organization processes in programming.

III. SYNTHESIS OPERATIONS

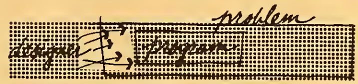
The operations performed in synthesis as a response to the program vary from designer to designer. They depend upon his VALUES as reflected in his VIEW OF DESIGN.



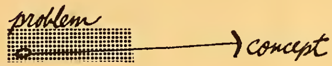
A. No matter how differently two designers may operate in synthesis, they are both essentially concerned with the CUMULATIVE establishment of relationships that will eventually result in a SINGULAR solution.



B. The way the designer "gets into" the problem is as important a DETERMINANT as the program data. The starting point for the designer may involve:



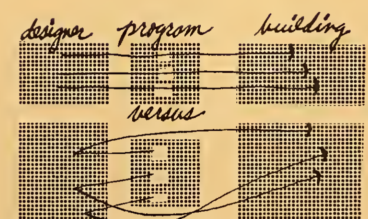
1. solving for CRITICAL issues
2. deriving an overall concept from the ESSENCE of the problem



3. working out the easy problems first and then the more difficult ones or vice versa
4. doing an OVERVIEW of the whole situation to establish relationships between major determinants
5. attending to the UNIQUE aspects of the problem before dealing with the more general or universal ones (pedestrian-car separation), or vice versa
6. searching for dimensional relationships between spaces and between spaces and the existing context for possible geometric organizational concepts

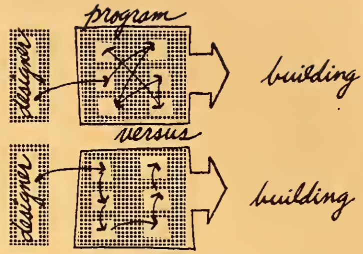
C. Some of the traditional issues related to synthesis as a CONTINUATION of programs are:

1. LITERAL RESPONSE VERSUS ARTISTIC RESPONSE. Depending on the designer's attitude about the nature of the facts and his ROLE in the design process he may attempt to make his design a literal translation of the program or an artistic expression of the program. The first of these views FACTS as crucial to the success of the building, while the second sees them as the basis for a creative INTERPRETATION.

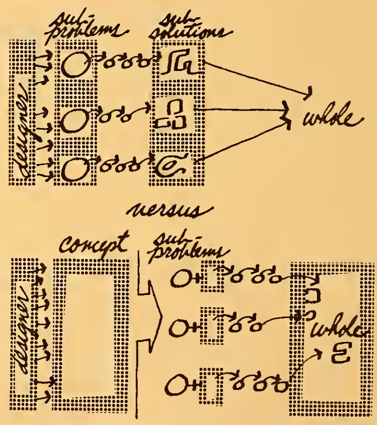


Related to the literal versus artistic response issue is that of INTERFACE between program and design. Synthesis may vary in its interface with programming both in terms of DEGREE (percentage of program having DIRECT relatedness to solution) and LEVEL (relative broadness or specificness of the program issues responded to directly).

2. RANDOM RESPONSE VERSUS STRUCTURED RESPONSE. The designer may give little thought to what part of the program he will attend to FIRST. "One point of beginning is as good as another." In contrast, a structured response requires a review of the program and then a PLAN for HOW it will be responded to in design. The structured response assumes that the SEQUENCE and MANNER OF designing from the program is a real influence on the nature and success of the final solution.

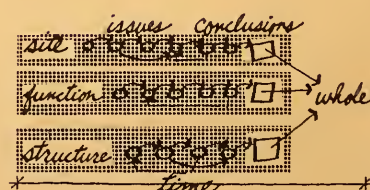


3. SUBOPTIMIZED APPROACH VERSUS CONCEPT FRAMEWORK APPROACH. The first of these entails ISOLATING specific problems and searching for solutions to them INDEPENDENTLY of each other. The solutions may involve the same architectural elements arranged different ways due to the different design criteria. The "optimal solutions" to these individual situations are then related to each other to make a WHOLE. This approach is very effective in pointing out the COMPETITION for form between the various problem determinants as the designer attempts to combine the sub-solutions without compromising them. The concept framework approach leads first to the generation of the "big idea(s)" or OVERALL organizational structure for the solution. The solutions to sub-problems then involve the ADDED determinant of "relating to the whole."

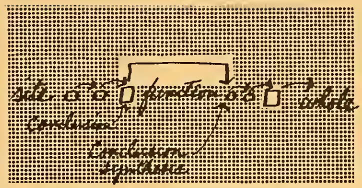


The first of these approaches values the attitude that the overall composition and sense of order should be derived from solving problems at the level where the building functions. "The building is a composite of solutions to individual problems." The second approach values a more controlled, ordered and structured sense of "whole." Compromises here are made in favor of the total rather than the part.

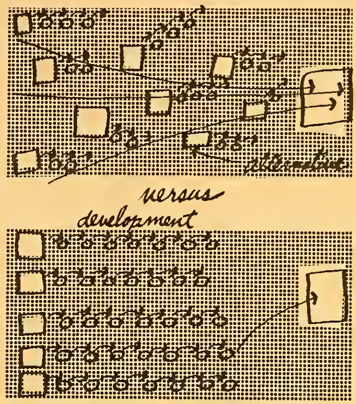
4. SIMULTANEOUS AND PARALLEL DEVELOPMENT OF ISSUES VERSUS SEQUENTIAL INTEGRATION OF ISSUES. The first of these pertains to working on different categories of the program SIMULTANEOUSLY but SEPARATELY (function and site). It is a form of suboptimization. Eventually conclusions



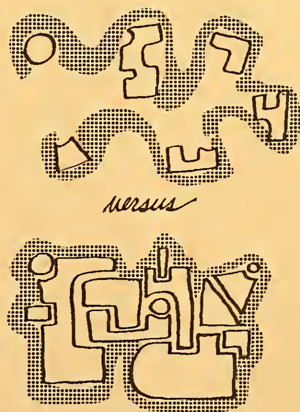
are drawn in each category and they are integrated. The second approach studies one topic until tentative conclusions are reached. Then another topic is studied together IN CONTEXT with the first. Conflicts are resolved and conclusions drawn about the synthesis of the two. The process continues until all the topics are covered. In this system, the sequence of topics studied is vital.



5. CONVERGENCE TO ONE SOLUTION VERSUS GENERATION OF ALTERNATIVES. Although alternatives are generated in the first of these, they are IMMEDIATELY judged and either discarded or incorporated into the solution. The approach values spending MINIMAL time in developing what will prove to be inviable alternatives. (One will be chosen and the others discarded.) The designer attempts to work for the solution that BEST responds to the program. He CONVERGES to that solution by making judgments about alternatives "as he goes" rather than by developing them and choosing later. The second viewpoint values the use of different solutions to help insure that the best direction will be taken in solving the problem by looking at the SPECTRUM of possibilities. These alternatives also serve as CATALYSTS for developing further concepts and as criteria for determining the most viable direction to take.



6. SEGREGATIVE SOLUTION VERSUS INTEGRATIVE SOLUTION. The segregative solution minimizes sub-solution compromises by SEPARATING the individually generated forms insofar as possible. This usually involves relating forms to a circulation framework but NOT to each other. This is especially advantageous where UNUSUAL forms are generated which would be difficult to physically relate to each other. It also allows the designer or designers to work on parts of the design independently of other parts. The segregative approach demands a strong UNITING system or element for finally assembling the sub-solutions. The integrative approach attempts to "weave" the form together so that there are as many MUTUAL relationships between the parts of the whole as possible (physically, dimensionally, structurally, mechanically). Because there is a greater degreee of "fit" needed between elements there is usually more COMPROMISE involved in achieving the fit.

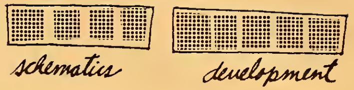


The first approach tends to generate an "assembly of differences" while the second results in a more "unified whole" where elements "belong" to each other.

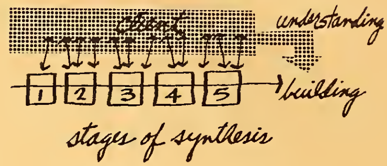
D. The designer's METHODS in synthesis are largely depen-

dent on his VIEW OF DESIGN. The models he uses for ordering the design situation are related to the models he uses for ordering his everyday experience.

E. The designer may divide synthesis into several stages. Even the DECOMPOSITION of schematics and design development into smaller increments may be necessary depending on the COMPLEXITY of the project and the FREQUENCY of needed client participation in the synthesis process.

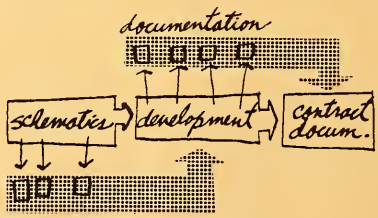


F. It is important that contact with the client be maintained through ALL stages of synthesis including the conceptual stages. Vital is the communication with the client in a manner which he can UNDERSTAND. This will help avoid the problem of the client not really knowing what his design is until it is built (with accompanying criticism, dissatisfaction and changes to a constructed building).



G. To successfully "take the client along" through the reasoning that leads to the physical architectural solution requires that the designer be highly ORGANIZED in his logical sequence of decision-making. The discipline of having to COMMUNICATE why you do what you do is an excellent test of PROBLEM ORGANIZATION.

H. Just as the RECORD KEEPING during the gathering process in programming can facilitate the other programming operations, the RECORD KEEPING during schematics can help during design development. Well ordered and documented schematic and development stages in turn can aid in executing the contract documents. Each stage in the entire process should ANTICIPATE and SIMULATE the following work.

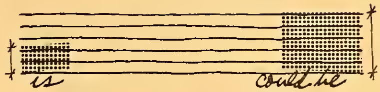


PROGRAM AND DESIGN EVALUATION

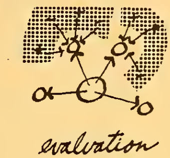
I. DEFINITIONS AND CONCEPTS

A. Evaluation: "An APPRAISAL of the VALUE or WORTH of something."

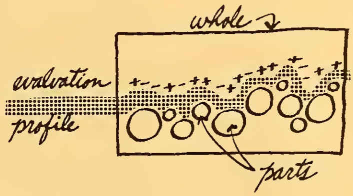
B. To evaluate something is to judge it against some STANDARD or SCALE. Evaluations always involve a RELATIONSHIP between what COULD be or SHOULD be and what IS.



C. EVALUATION is distinct from ANALYSIS in that evaluation involves a VALUE JUDGMENT (not necessarily in the subjective sense). Analysis is concerned only with the decomposition of a whole into its parts. Evaluation may be preceded by analysis, but analysis doesn't necessarily require an evaluation. The one is DESCRIPTIVE while the other is EVALUATIVE.



D. Evaluation can occur at varying levels of generality or specificity. We can appraise a WHOLE or its PARTS. Because it is desirable for there to be a close "fit" between the "evaluation profile" and the profile of the thing being evaluated, it seems best if INDIVIDUAL judgments are made about specific COMPONENT aspects of the "whole." The cumulative judgment of the parts IS the judgment of the whole.



In the same way that a "whole" cannot be designed as such but results from attention paid to the relationships of comprising parts, so also it seems meaningless to attempt to evaluate the "whole." Even so-called immediate, over-all, general positive or negative responses to things are based on SPECIFIC qualities such as visual appeal.

Individual parts may be judged by totally DIFFERENT criteria in evaluation.

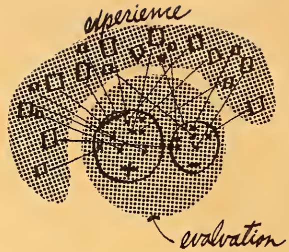
E. The model of "ordering systems" serves as a useful means for understanding the EVALUATION process just as it helps in the understanding of DESIGN SYNTHESIS.

Evaluations are made NOT of the objects or things themselves but of their QUALITIES. The same scalar characteristic of properties of elements which is used for ordering the elements in design is used in evaluation.

F. Properties of elements and relationships are judged in a manner which is essentially QUANTITATIVE. The degree to which the object to be evaluated possesses the desired

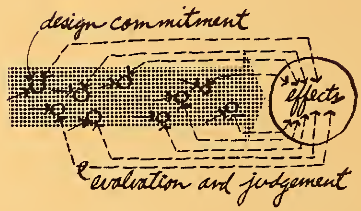
quality or qualities determines the extent to which we VALUE it. Even though the choice of the quality to be used for the evaluation may be subjective, once selected, the judgment may be made quantitatively.

G. The concept of evaluation is rooted in the model of the necessity of gratifying self-love. We ASSIGN positivity or negativity to experience based on its PERCEIVED affect on our own SELF-ESTEEM. We attend to phenomena only when they potentially may be of CONSEQUENCE in some way to our self-concept (which may range from physical well-being to psychological considerations). We are most SENSITIVELY attentive to those things which we have grown to be most DEPENDENT upon for gratification of self-love. Once attended to, experience is "evaluated" or categorized in terms of its relative SUPPORT of or DAMAGE to our self-image.



II. EVALUATION IN PROGRAMMING & DESIGN

A. Although normally we may think of "evaluation" as applied to final designs and finished buildings, its role extends through the entire programming and design process. We are continually making TENTATIVE COMMITMENTS and judging them against DESIRED CONSEQUENCES or goals. The criteria for making these "sub-evaluations" are as numerous as those used to make the commitments (visual, functional, mechanical, structural, sensory).



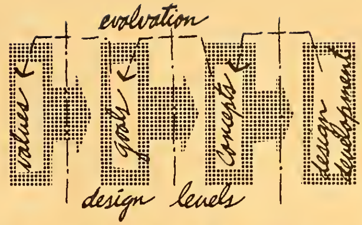
B. Examples of evaluations made at all stages of the planning process follow. Evaluation in programming and design is an ESSENTIAL aspect of the decision-making process as well as simply the process of making sense (ordering) of experience.

1. Will the client be easy to work with?
2. Should you accept the job?
3. Is the commission socially significant?
4. What tactic would be best for data gathering?
5. How best can the information be organized?
6. What are the most important issues?
7. What value will you assign the various data?
8. Which alternative concepts should be pursued further?
9. Which concept is most viable?
10. How should the working drawings be structured?
11. Do you want open bidding or bidding by invitation?
12. Was the building successful?
13. Was the job successful?

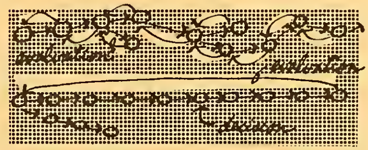
C. Evaluation requires that there be a desired GOAL or STANDARD and a commitment to judge against that

standard. Evaluation may be in terms of "unspoken" criteria such as economy of effort or in terms of criteria which are EXPRESSED.

D. The evaluation process may occur at ANY LEVEL from the very GENERAL to the very PARTICULAR in programming and design. The smallest design development decision may be evaluated within the FRAMEWORK of the broader project CONCEPT. CONCEPTS may be judged against the problem GOALS. Problem GOALS may be evaluated in the light of the VALUES of the programmer or designer and how his values relate to the issue of satisfying the client's needs and wants.



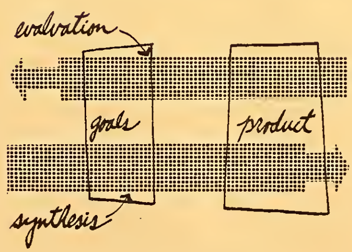
E. Some of the evaluations made in programming and design offer FEEDBACK immediately to decisions and affect the process "en route," while others are made only AFTER more complex and lengthy decision-making processes have been completed. (Evaluate the building to determine whether the whole process needs to be recycled.)



F. Evaluation as a task becomes more difficult when goals have not been EXPLICITLY set PRIOR to proceeding with programming and design. The more declarative and specific the goals, the EASIER the task of evaluation.

G. Evaluation on the basis of ARBITRARILY determined criteria is as much of a problem as design on the basis of arbitrarily determined criteria.

H. We can think of the evaluative process as design synthesis in REVERSE. SYNTHESIS proceeds from goals to product while evaluation proceeds from product to goals. In synthesis we may think of the problem in terms of two major concerns: PHILOSOPHY – GOALS – ASSUMPTIONS and CONSISTENCY OF EXECUTION. Evaluation of a project may be made in terms of these same two aspects. For example, a project may be VALID as to its goals but INCONSISTENTLY executed or may be a magnificently CONSISTENT execution of an INVALID assumption.



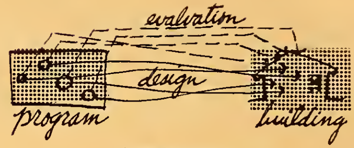
I. An evaluation may be based on SUBCONSCIOUS criteria known by experience or a careful and systematic evaluation on the basis of logically constructed criteria generated by CONSCIOUS thought and recorded verbally and graphically.

J. Using the same design model as mentioned in the Introduction, the evaluation of the final product of the design process should be based on the EXTENT to which it generated the desired CONSEQUENCES. Evaluation of a design prior to construction must necessarily be based on past experience of cause-effect relationships between physical FORM and

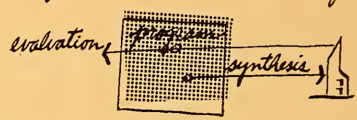
resulting CONSEQUENCES. It is not unreasonable to expect that observed consequences or hypothetical consequences may be highly positive but not PROJECTED or EXPECTED at the beginning of the planning process. This serendipity may sometimes prompt a re-evaluation of the originally stated goals and desired consequences.

III. PROGRAM AS AN EVALUATIVE TOOL

A. The EVALUATION of completed buildings is a rapidly developing ASPECT of architectural design. This is evidenced by the inclusion of this specialty into the curricula of many architectural graduate schools.



B. It would seem appropriate in the evaluation of a building (or unbuilt design) that it be judged in the light of the INTENTIONS that formed it. These intentions are probably nowhere as CLEARLY stated as in the building program.



C. Although the principle role of the architectural program is that of a DIRECTION GIVING device in synthesis, it may also serve as a tool for EVALUATING the final design solution.

D. The FORM of the program should facilitate its use as an EVALUATIVE tool. Critical issues should be identified and precepts should be CONCISE and DECLARATIVE. It becomes much easier to judge the relative success of a building when the program states what SHOULD HAPPEN in the building.

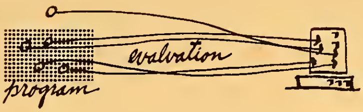
E. In its ideal form, the evaluative situation should involve a program that PREDICTS desired EFFECTS and CONSEQUENCES and an observation to determine if these consequences do in fact OCCUR and if they are indeed DESIRABLE.

F. The program "sets the tone" for the evaluation. A well documented systematic program will usually prompt a THOROUGH and well ORGANIZED evaluation.

G. The use of the program to evaluate the building can serve as an INDIRECT EVALUATION of the PROGRAM itself.

1. If the program is of LITTLE help in evaluating the building, it was probably of LITTLE help in the design of the building.

2. Critical issues which are BURIED in supporting data present a problem to the evaluator and designer alike.



3. When the evaluation deals with points and issues NOT COVERED in the program, this is an indication that the program may not have been COMPLETE or THOROUGH.

4. A good program format for EVALUATING a design is also a good one to DESIGN from.

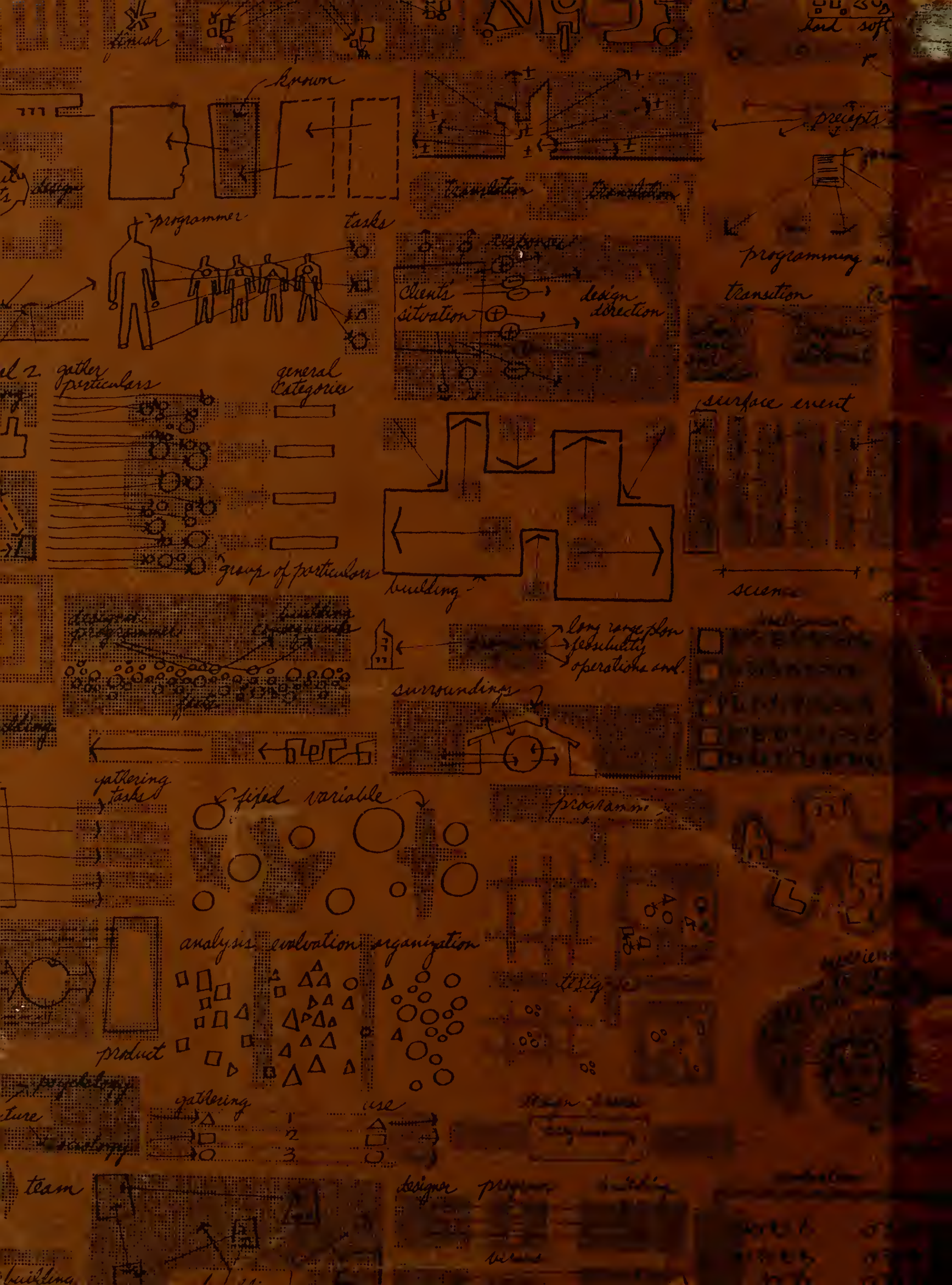
H. The FORMAT and organization of the evaluation may be related but not limited to the FORMAT of the program.

I. A reorganized TABLE OF CONTENTS for the evaluation may be taken as a suggested improvement in the PROGRAM table of contents.

[redacted]

Date Due

Due	Returned	Due	Returned
[illegible]			
JUN 2 2 '78			
OCT 2 6 '78			
OCT 3 0 '78			
MAY 2 2 '80			
[illegible]			
SEP 1 6 [illegible]			
OCT 2 3 1984			
DEC 1 8 1984			
NOV 0 7 1988	NOV 1 5 1988		
JAN 2 9 1990	JAN 2 4 1990		
DEC 0 6 2005	DEC 1 9 2005		


finish
known
precepts
programmer
tasks
clients
situation
design
direction
programming
transition
gather
particulars
general
categories
surface event
group of particulars
building
science
long range plan
feasibility
operations and.
surroundings
gathering
tasks
fixed
variable
analysis
evaluation
organization
product
gathering
use
1
2
3
team
designer